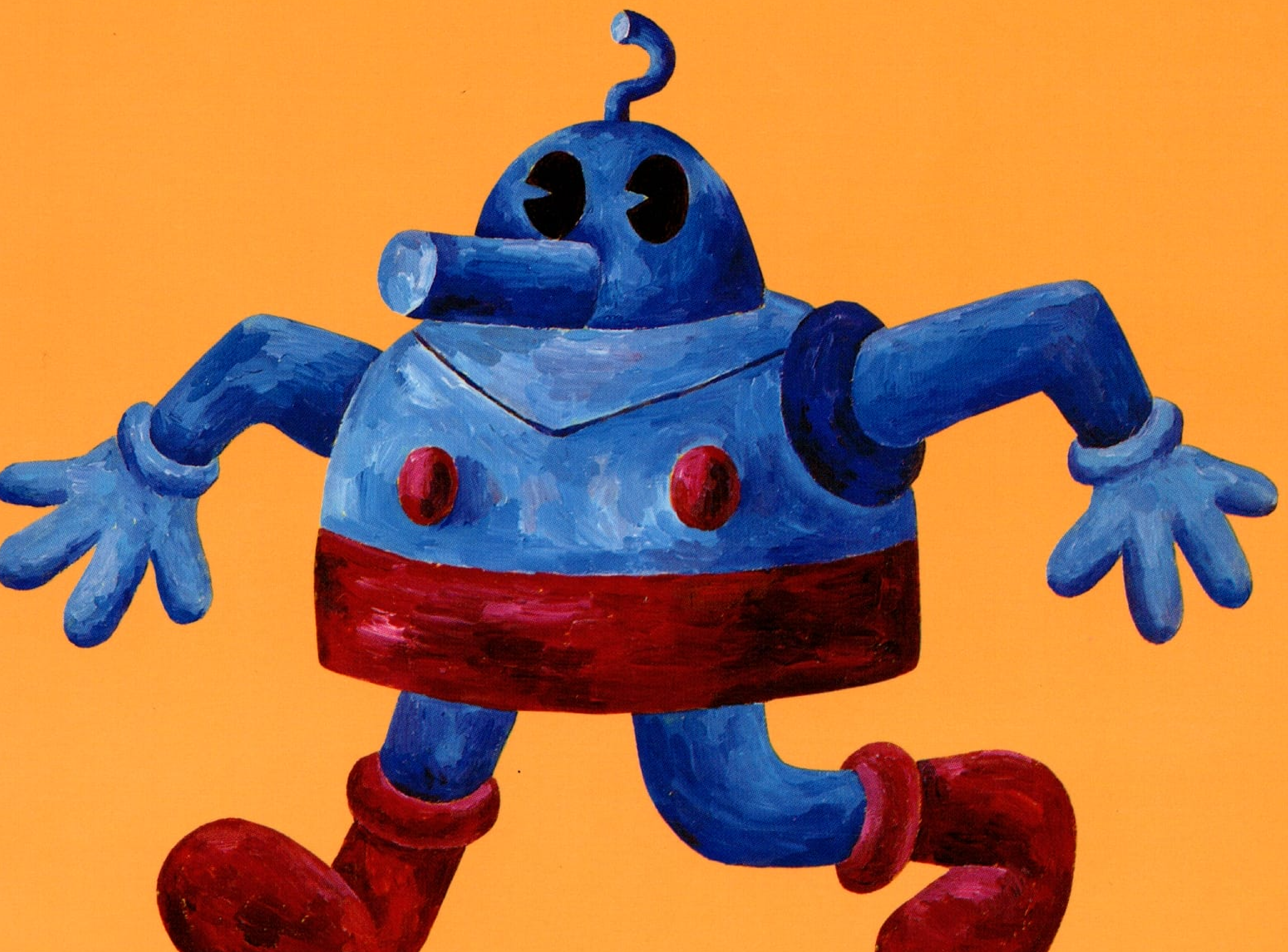
MSX POCKET BANK 1

ポケットバンク

アニメC.G.(コンピュータ・グラフィック)に挑戦!

川野名 勇・牧山慶士



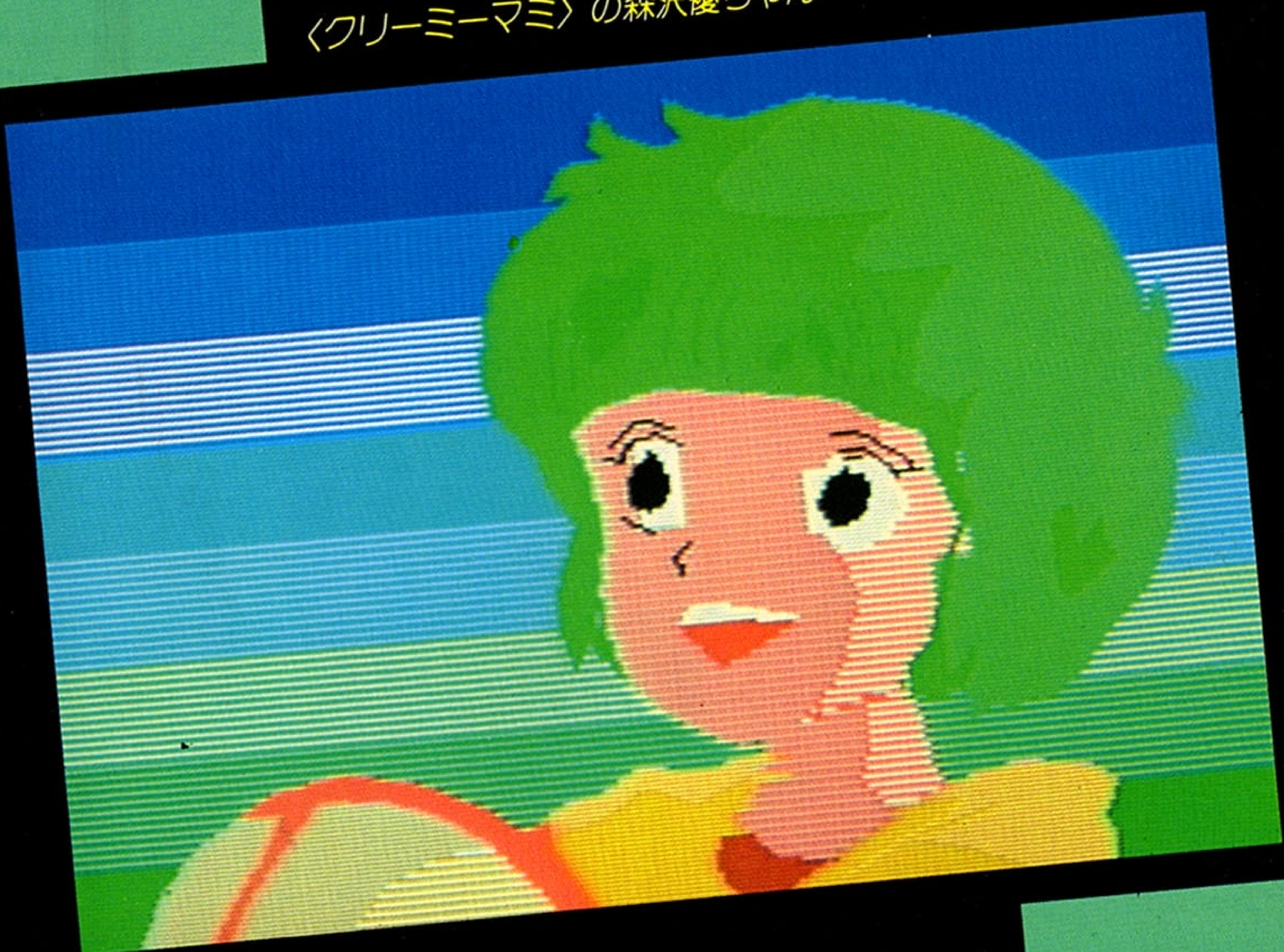
アニメを描きたくなった……
マイコンをスケッチブックに変えると
マミがウィンクした

アスキー出版局

C.G. Character

アニメ・キャラクタに挑戦しよう！

〈クリーミーマミ〉の森沢優ちゃん



C.G. Character

C.G. Character

〈The かぼちゃワイン〉のエルで〜す



© 三浦みつる・テレビ朝日・東映動画

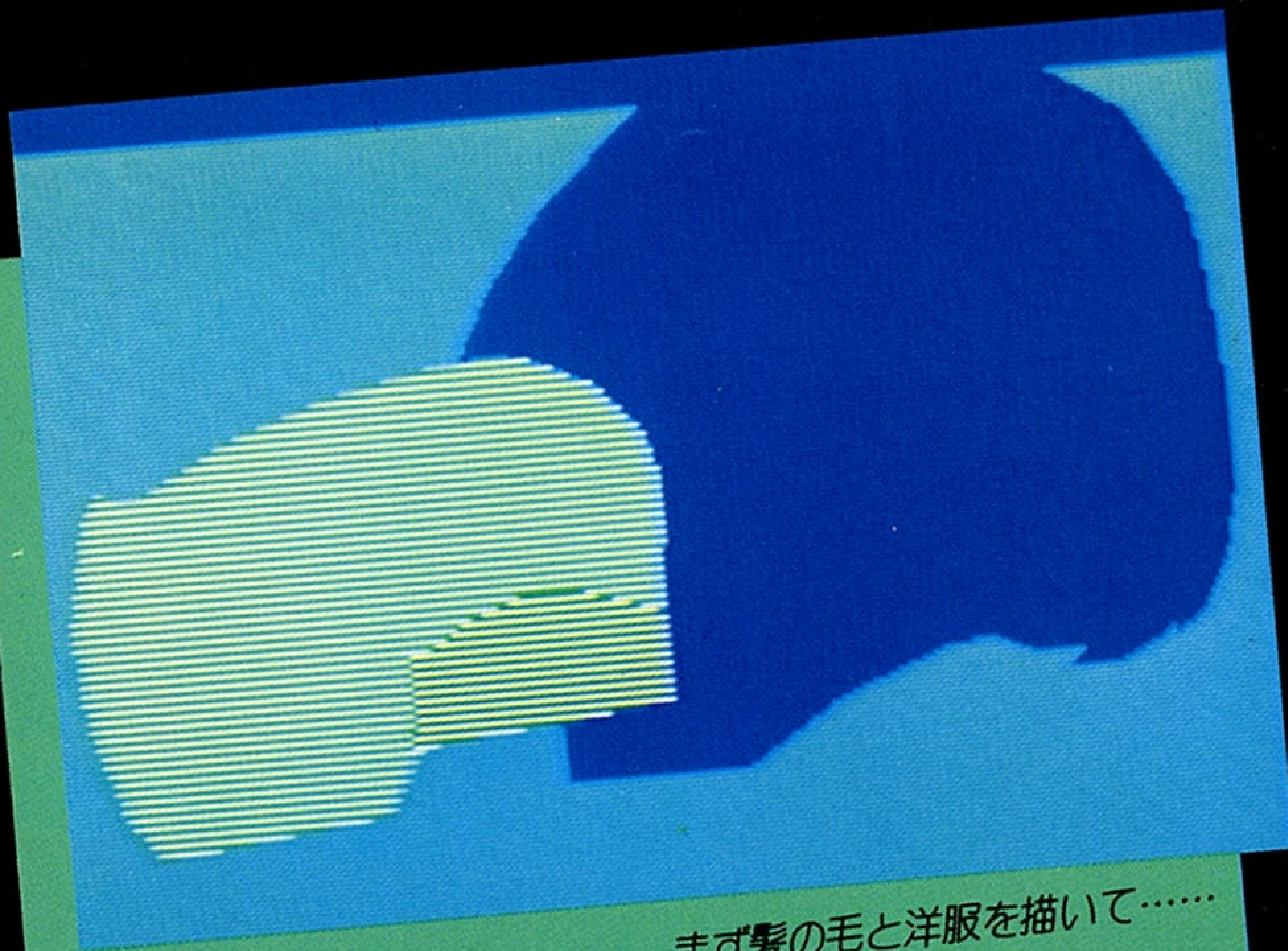
〈ミンキーモモ〉忘れないでね！



© 葦プロダクション

C.G. Character

一枚の絵ができるまで……



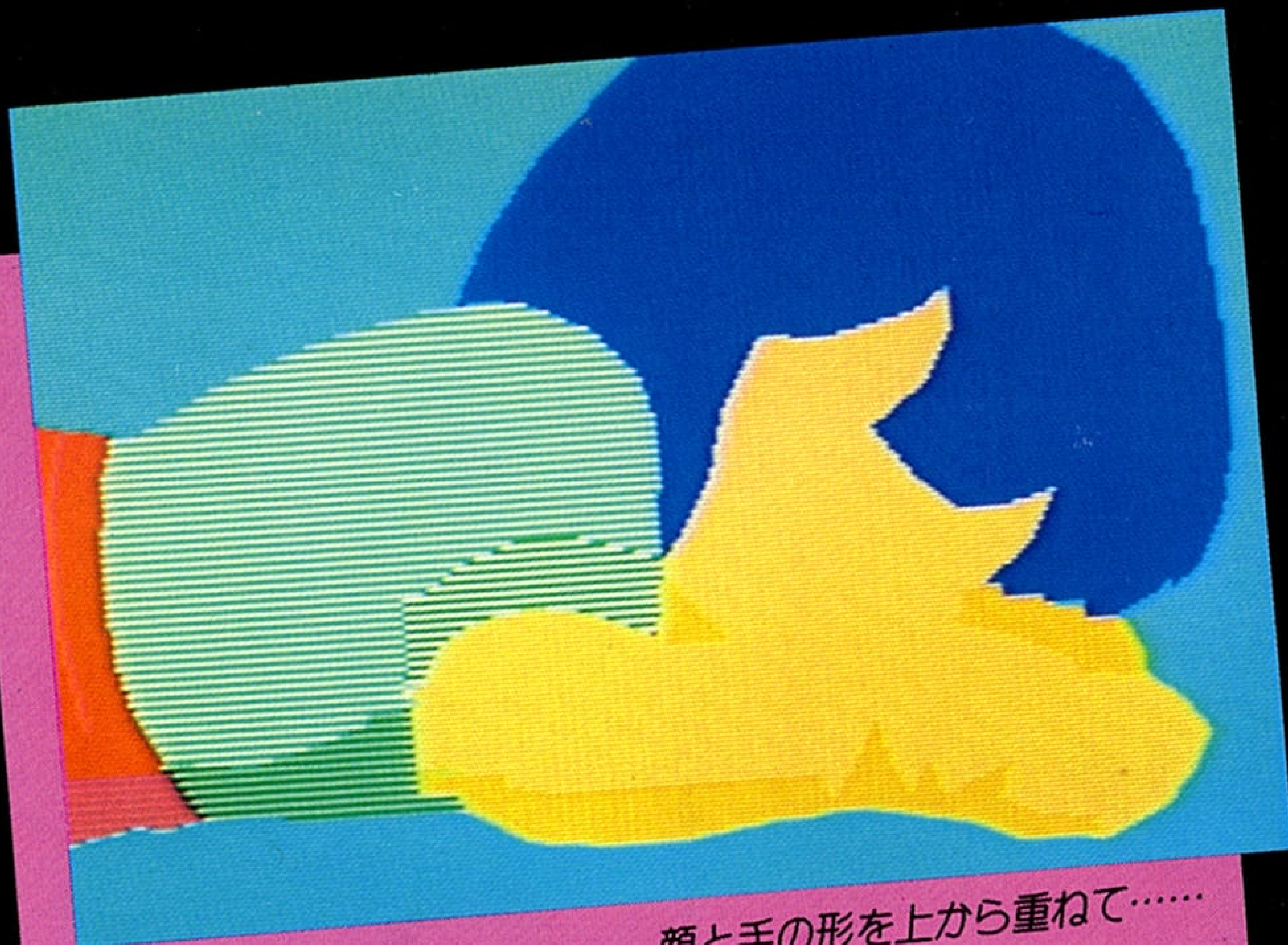
① まず髪の毛と洋服を描いて……



② 次にスカートを描きます

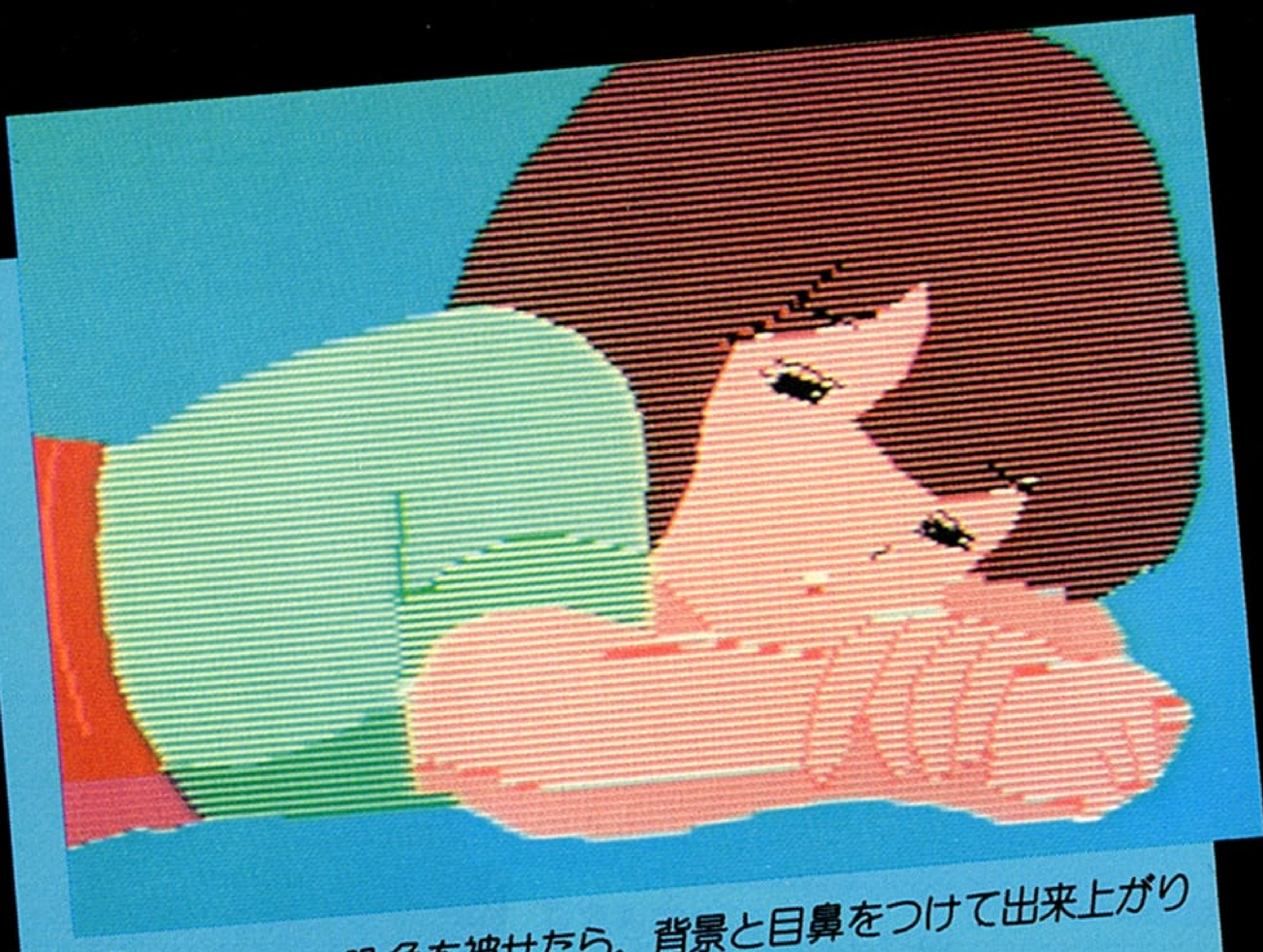
C.G. Character

（新潟大学佐藤くんのオリジナル・キャラクタです）



顔と手の形を上から重ねて……

③



肌色を被せたら，背景と目鼻をつけて出来上がり

④

C.G. Character

MSX POCKET BANK 1

コンピュータ・グラフィック

アニメC.G.に挑戦!

川野名 勇
牧山慶士

アスキー出版局

この本は，MSX パソコンで「お絵描き」をしてみようという人のために作られたものだ．「お絵描き」といっても，ただの絵ではない．ちかごろ，テレビのブラウン管の中だけでは満足できずに，映画館のスクリーンでも大活躍している，あのあこがれのアニメ・ヒーローたちを，パソコンのディスプレイ画面に描いてしまおうというのである．

コンピュータ・グラフィクス（C. G.）なるものが話題になりだしてから，ずいぶんたつが，映像そのものは，もうテレビや映画ですっかりおなじみになった．その間，われらがパソコンのほうの表示能力もぐっとアップし，安価な機種でもカラフルできめ細かい絵が描けるようになっている．MSX パソコンだって，256×192ドット，16色までの表示が可能だ．

こうしたグラフィック能力を備えたパソコンにとって，単純な線とベタ塗りで構成されたカラフルなアニメ主人公たちは，まさにうってつけの題材だ．本書では，グラフィック命令のイロハからはじめて，パソコンでアニメ・キャラクタを描く方法を述べていく．

後半ではかなり本格的なグラフィック・ツールを紹介する．また，描き上げたキャラクタのデータも巻末に載せておいた．

ともかく MSX にプログラムを打ちこもう．そして，グラフィック・ツールで絵を描いてみよう．

ディスプレイの上に生き生きとしたアニメ・キャラクタたちを再現するには，油絵なみの粘り強さと，水彩画なみのしゃれたセンスを必要とすることがわかると思う．

もし，オリジナルの素晴らしい絵ができ上がったら，ぜひ編集部あてに送ってほしい．では，アニメ C. G.の第一歩を踏み出そう．

INTRODUCTION

表紙デザイン　HIT STUDIO
表紙イラストレーション　スージー甘金
本文イラストレーション　汲田幸英

協力：新潟大学コンピュータ同好会

CONTENTS

ONE

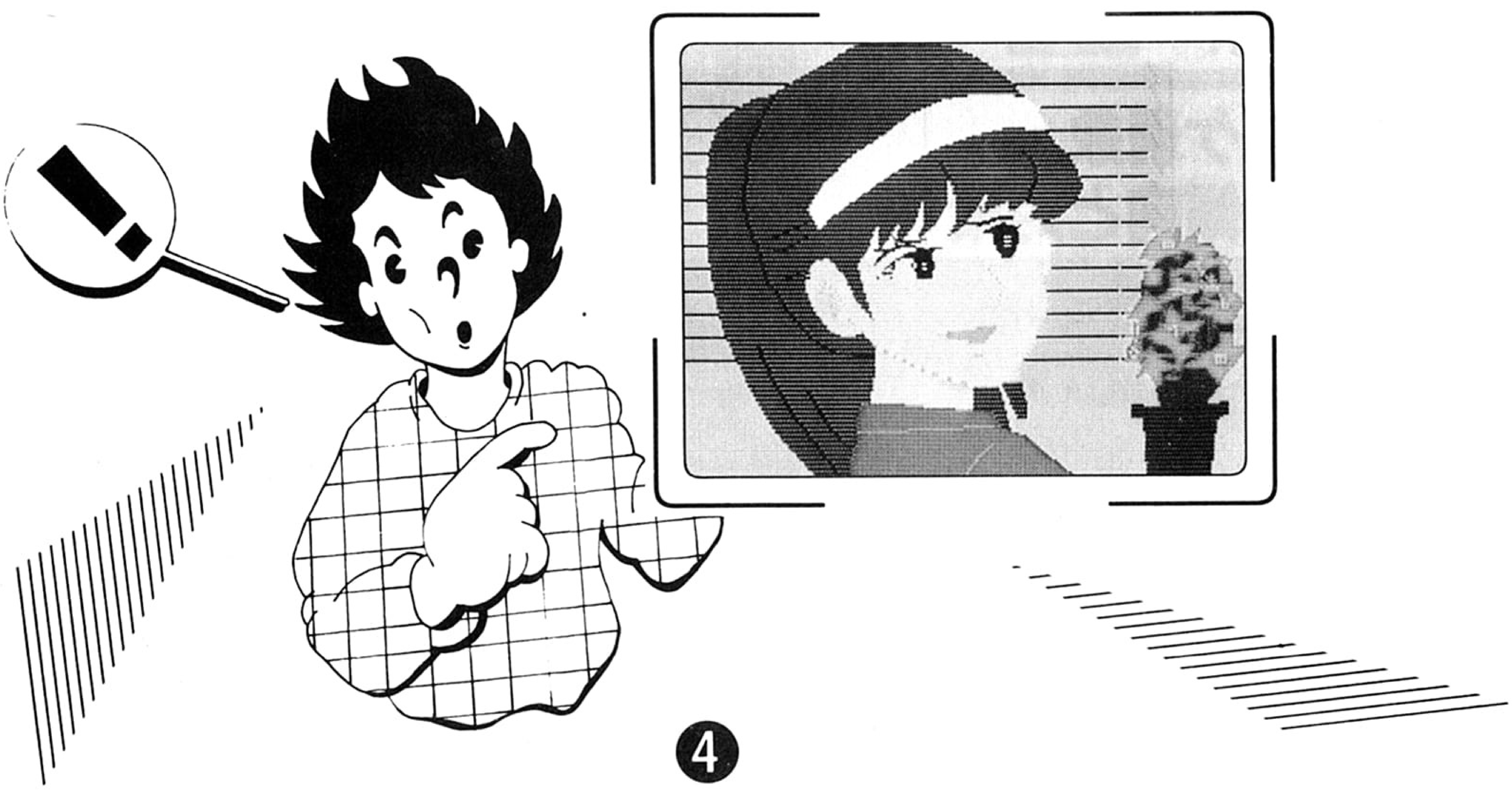
Part 1

Let's get C.G.

グラフィックの基礎知識

さて，一刻も早くあこがれのアニメ主人公たちをディスプレイの上に登場させたい，とウズウズしている人たちには申しわけないのだけれども，その前にまずパソコンで絵や図形を描くために必要な，ごく基本的な知識についてすこし．

何ごとも基本が大切——などというと，どこかの野球部のカントクのようで大げさになってしまうが，パソコンを使って絵を描くためにも，やはり知っておくべきことがいくつかある．ウォーミング・アップなしでいきなり大作に取り組んでも，途中で息切れするのがオチだし，だいいち基本ができていないと応用もきかない．ここで覚えることはアニメ主人公以外のもの（グラフやゲームの背景など）を描くときにも，きっと役に立つはずだ．いわば本格的アニメ・キャラクタの作画にとりかかる前の，パソコン絵画教室入門編である．気軽に読んでいただきたい．



1.1 数字で描くパソコン絵画

私たちが紙の上に絵や図形を描くときには，鉛筆を握った手を動かし，何本もの線を描いたり消したりしていくのがふつうだ．そして，おおまかな輪郭ができ上がったところで，絵筆で色をつけていく．もちろん，直線や円を描くのに，フリーハンドではなく定規やコンパスを使うこともある．

この手順は，パソコンで絵を描く場合にもほぼ同じだ．ただ，パソコンの場合には，こうした絵を描く動作のひとつひとつを，命令をとおしてコンピュータに指示してやる必要がある．

ものは試し，画面に線を一本引いてみることにしよう．次のリストを打ちこんで，RUN(ラン)してみてほしい．

```
10 SCREEN 2
20 LINE(0,0)-(255,191),15
30 GOTO 30
```

実行と共に，画面の左上から右下にかけて，直線が描かれたはずだ．くわしい説明はあとにまわすが，このリストの中で，20行が「画面の（0，0）の地点から（255，191）の地点まで，15番の色で直線を引きなさい」という命令になっている．このように，パソコンに線を描かせるには，どのような線をどの位置に描くかということを，いちいち数字で指定してやらねばならない．数字で指示するところがいかにもコンピュータ的ともいえるだろう．

なお，リストの30行目を見てもらえばわかると思うが，このプログラムは一度実行させるとそのままでは止まらないようになっている．END(エンド)文を使って止めるようにすると，線を描いたとたんにまた消えてしまうからだ．

プログラムを止めるには，ご存じとも思うが，[CTRL]キーを押しながら[STOP]キーを押す．本書で今後説明のために出てくるプログラムは，ほとんど自動的に止まるようになっていないから，この止め方は忘れないでほしい．

MSXパソコンをはじめとして，多くのパソコンのBASICには，いまあげたLINE(ライン)命令（直線を描く）のほか，CIRCLE(サークル)命令（円を描く），PAINT(ペイント)命令（決

められた区域を塗りつぶす）などの便利な命令が用意されている．これらの，絵や図形を描くための一群の命令のことを，グラフィック命令と呼ぶ．雑誌などによく載っている「ユウちゃん」の絵なども，基本的にはグラフィック命令をいくつも組み合わせていくことで描かれるわけだ．グラフィック命令とは，パソコンで絵を描くための三角定規やコンパスにあたると考えればよいだろう．

それでは，個々のグラフィック命令についてざっと見ていくことにしよう．

1.2 6つの基本命令

■画面モードを決めるSCREEN命令

まず，絵を描くためには画用紙を用意しなければならない．パソコンの場合には，画面をグラフィック・モード（絵を描くためのモード）に設定する必要がある．この設定を行うのが，さきほどのリストの10行に登場した「SCREEN（スクリーン）」という命令だ(SCREEN 命令には他にもいくつかの機能があるが，ここでは直接関係ないので説明を省く）．

MSX パソコンには，**表 1** のように 0 から 3 までの 4 つの画面モードが用意されている．モードの設定は，

SCREEN 〔画面モード〕

という形で行う．

画面モード	表示能力	プログラム・文字入力	グラフィック命令
0	40字×24行	○	×
1	32字×24行	○	×
2	256×192ドット	×	○
3	64×48ブロック	×	○

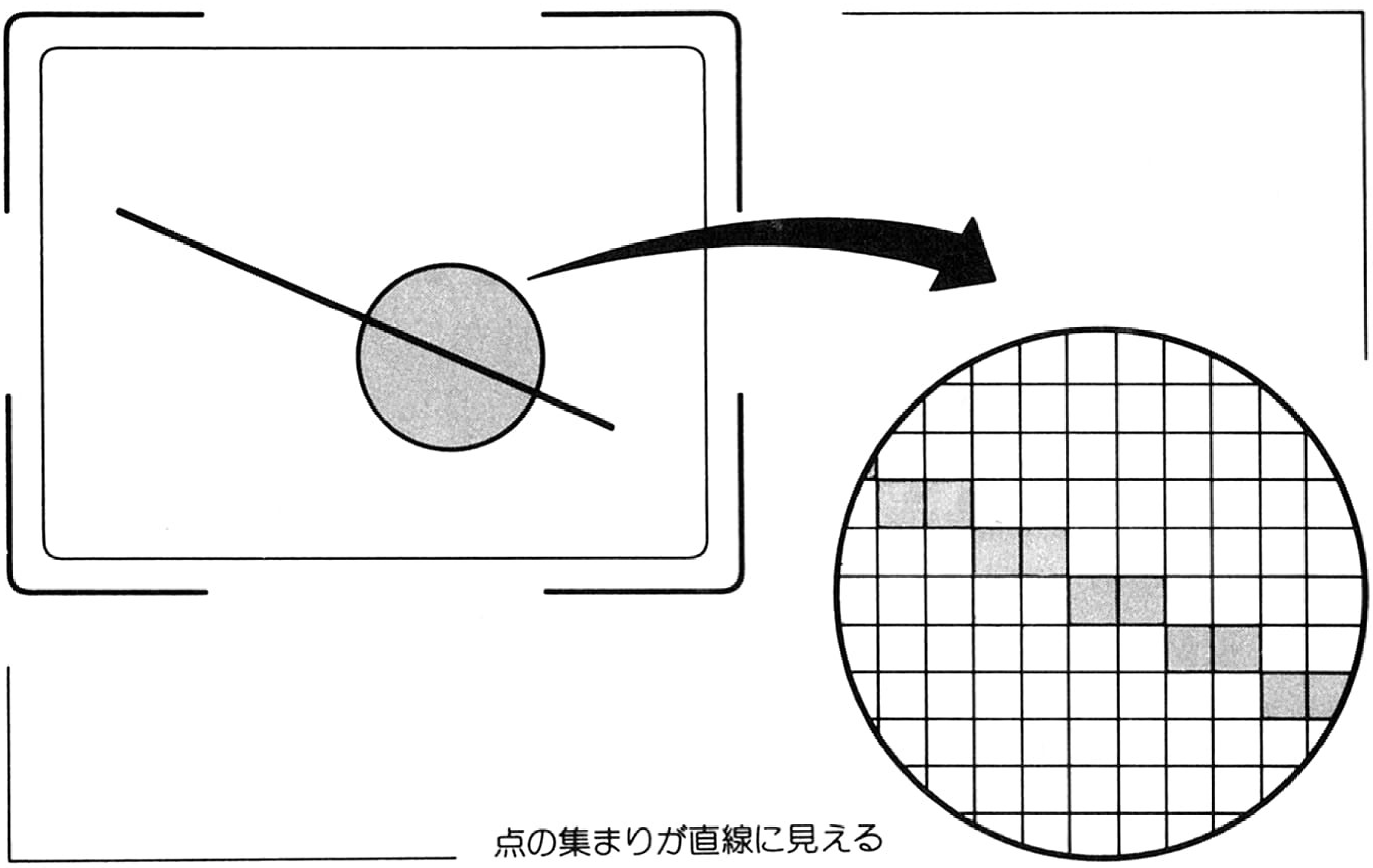
表 1

このうち，画面モードの０と１は，プログラムを入力したり，表示したりするためのもので，テキスト・モードと呼ばれる．このモードは文字表示用であるため，さきほど述べたグラフィック命令は使えない．

これに対して，モード２と３は，絵を描くためのモードだ．まず，画面モード２について説明していこう．

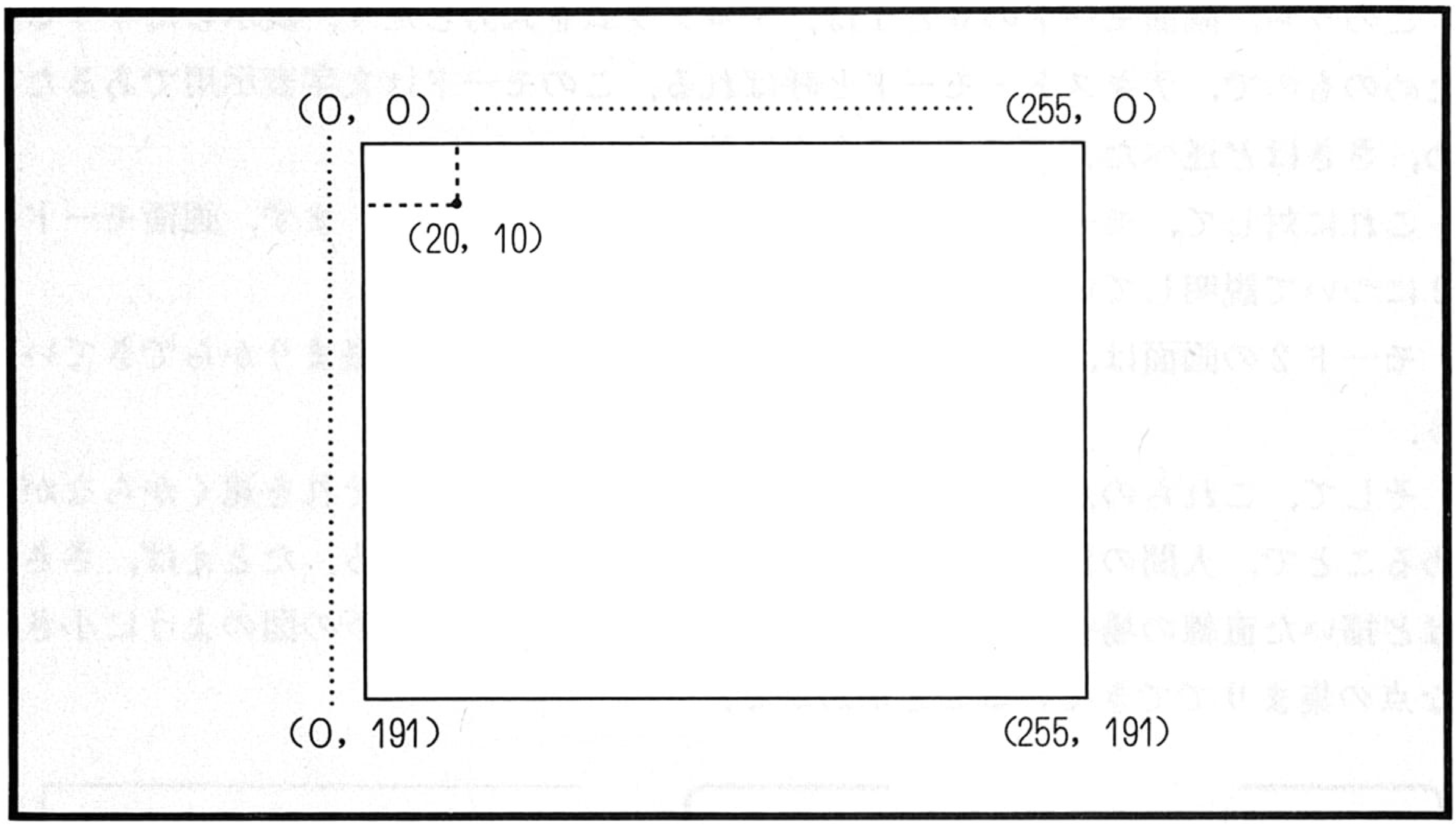
モード２の画面は，横256×縦192の細かい点（ドット）の集まりからできている．

そして，これらの点のひとつひとつに色がつけられると，それを遠くからながめることで，人間の目にはいろいろな図形や絵となって見える．たとえば，さきほど描いた直線の場合にも，線の一部分を拡大してみると，下の図のように小さな点の集まりでできていることがわかる．



点の集まりが直線に見える

また，ひとつひとつの点には，画面の一番左上端の点を（０，０）として，画面上の位置を表わす座標が決められている．たとえば，点（20，10）の位置は，画面左上端から少し斜め下にさがった地点を表わすわけだ．LINE，CIRCLE などのグラフィック命令で，「どこそこの地点」と指示を出す場合には，すべてこの座標を使って表わす（次ページ図参照）．



それでは，さきほど打ちこんだリストに，25行目として次の一行をつけ加えてみよう．

```
25 LINE(0,191)-(255,0),15
```

今度は，画面の上に大きな×印が描かれた．20行と25行をくらべてみると，直線を引く位置を指定する数値が変わっていることに気づく．このように，座標の概念は画面上に絵を描くときには大変重要なので，しっかりと身につけておいてほしい．

さて，モード2の話が長くなったが，モード3についてもすこし触れておこう．モード3も絵を描くためのものだが，画面を構成する点の大きさが，横64×縦48とかなり荒くなってしまう．直線を引く場合にも太くなり，斜めの線もギザギザが目立ってしまうわけだ．

今後本書では，アニメキャラクタなどの細かい絵を描いていくので，画面モードは2を中心に使う．ただ，画面モード2の場合には，色のつけかたに多少の制限がある．このあたりの点については，後の章でくわしく説明することにしよう．

■色を決めるカラーコード

画面上の位置を表わすのに座標を使うことはすでに述べたが，パソコンでは色を表わすのにも数字を使う．MSX パソコンは，全部で16色を表示することができるが，それぞれの色には，**表 2** のようなカラーコードと呼ばれる数値が決められている．

表 2

0	➡ 透明	8	➡ 赤
1	➡ 黒	9	➡ 明るい赤
2	➡ 緑	10	➡ 暗い黄色
3	➡ 明るい緑	11	➡ 明るい黄色
4	➡ 暗い青	12	➡ 暗い緑
5	➡ 明るい青	13	➡ マゼンタ
6	➡ 暗い赤	14	➡ 灰色
7	➡ シアン	15	➡ 白

［カラーコードと色の対応］

ここでもう一度，直線を引く命令を思い出してみよう．25行目の，

```
25 LINE(0,191)-(255,0),15
```

という命令文で，おしまいの数値15が，直線の色を決めている．この15という数値を 0 ～15の範囲でいろいろ変えてみると，引かれる直線の色が変わる．

それでは MSX パソコンで使える16色全部を表示させるプログラムを紹介しておこう．実際に自分の目で確かめてみてほしい．なお，打ちこむときには，NEW（ニュー）命令で前に打ちこんだプログラムをいったん消すことを忘れないように．

```
10 SCREEN 2
20 FOR I=0 TO 15
30 LINE(I*16,0)-((I+1)*16-1,191),I,BF
40 NEXT I
50 GOTO 50
```

■背景色を決めるCOLOR命令

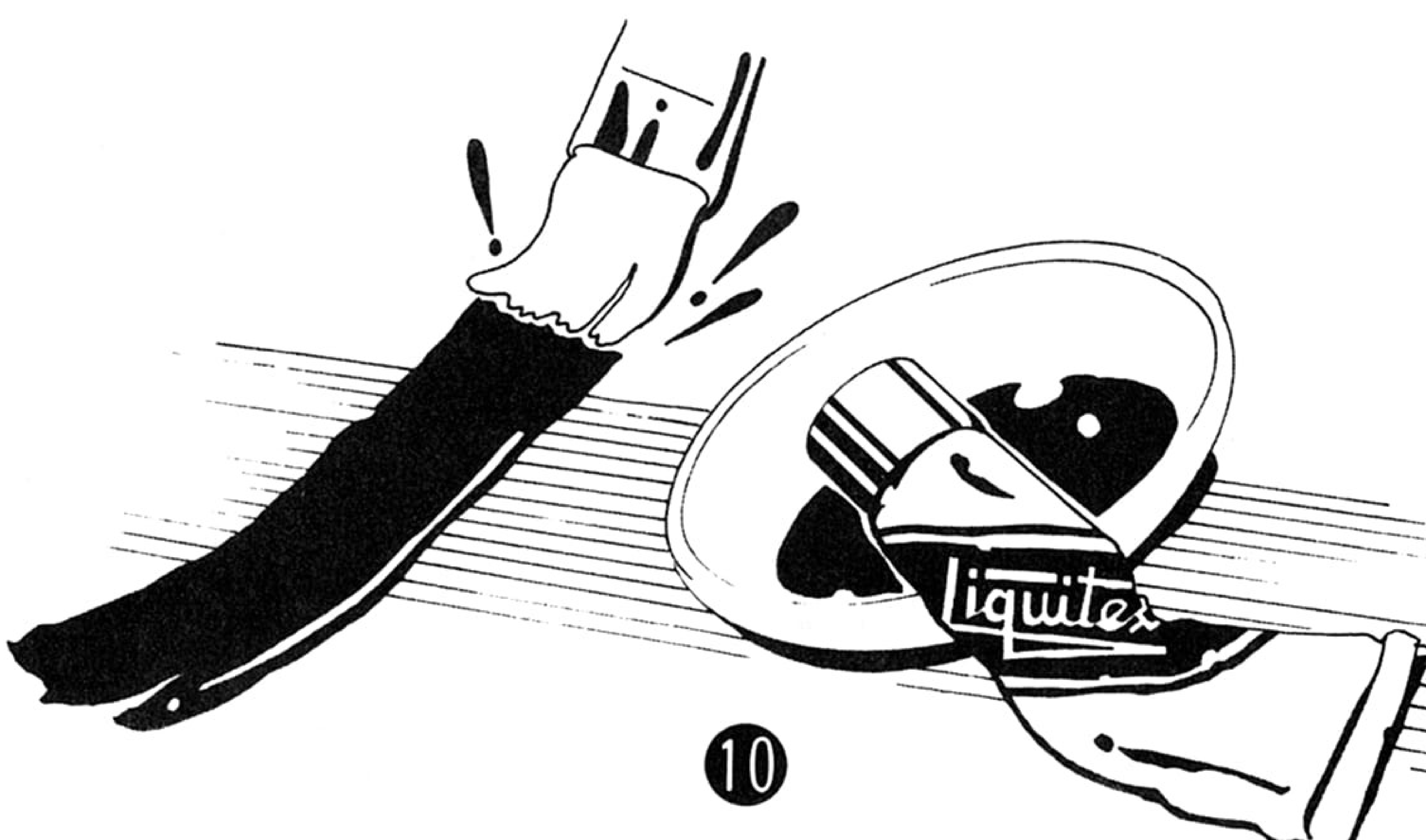
MSXのスイッチを入れたときには，画面の色は，中央の表示領域が青，外枠の部分がシアン，文字の色が白になっている．この3つの色は「COLOR(カラー)」という命令で変更することができる．次のプログラムでは表示領域と外枠の色を次々に変えている．

```
10 SCREEN 2
20 FOR I=0 TO 15
30 J=(I+4) MOD 15
40 COLOR 15,J,I:CLS
50 FOR W=0 TO 200:NEXT W
60 NEXT I
70 GOTO 70
```

COLOR命令の使い方は次のとおりだ．

COLOR 〔文字の色〕，〔表示領域の色〕，〔外枠の色〕

3つの色は，もちろんカラーコードで指示する．これであらかじめ，画面の背景となる色を決めて，「お絵描き」に備えることができる．ただし，画面モードが2の場合，表示領域の色は，COLOR命令を実行しただけではすぐに変化せず，続けてCLSという命令（画面に出ている文字や図形をすべて消し去る命令）を実行してやる必要があるので注意しよう．



■点を打つPSET命令

PSET(ピーセット)は画面上にポツリとひとつ点を打つ命令だ.

PSET(よこの座標, たての座標), 〔点の色〕

という形式で使う. ディスプレイによっては色のにじみのために, 指定した色が表示されない場合もあるが, しかたのないところだろう. PSET命令を使ってランダムに点を打つプログラムを紹介しておこう.

```
10 SCREEN 2
20 COLOR 15,1,7:CLS
30 X=RND(1)*255
40 Y=RND(1)*191
50 C=RND(1)*15
60 PSET(X,Y),C
70 FOR W=1 TO 100:NEXT W
80 GOTO 30
```

■直線を引くLINE命令

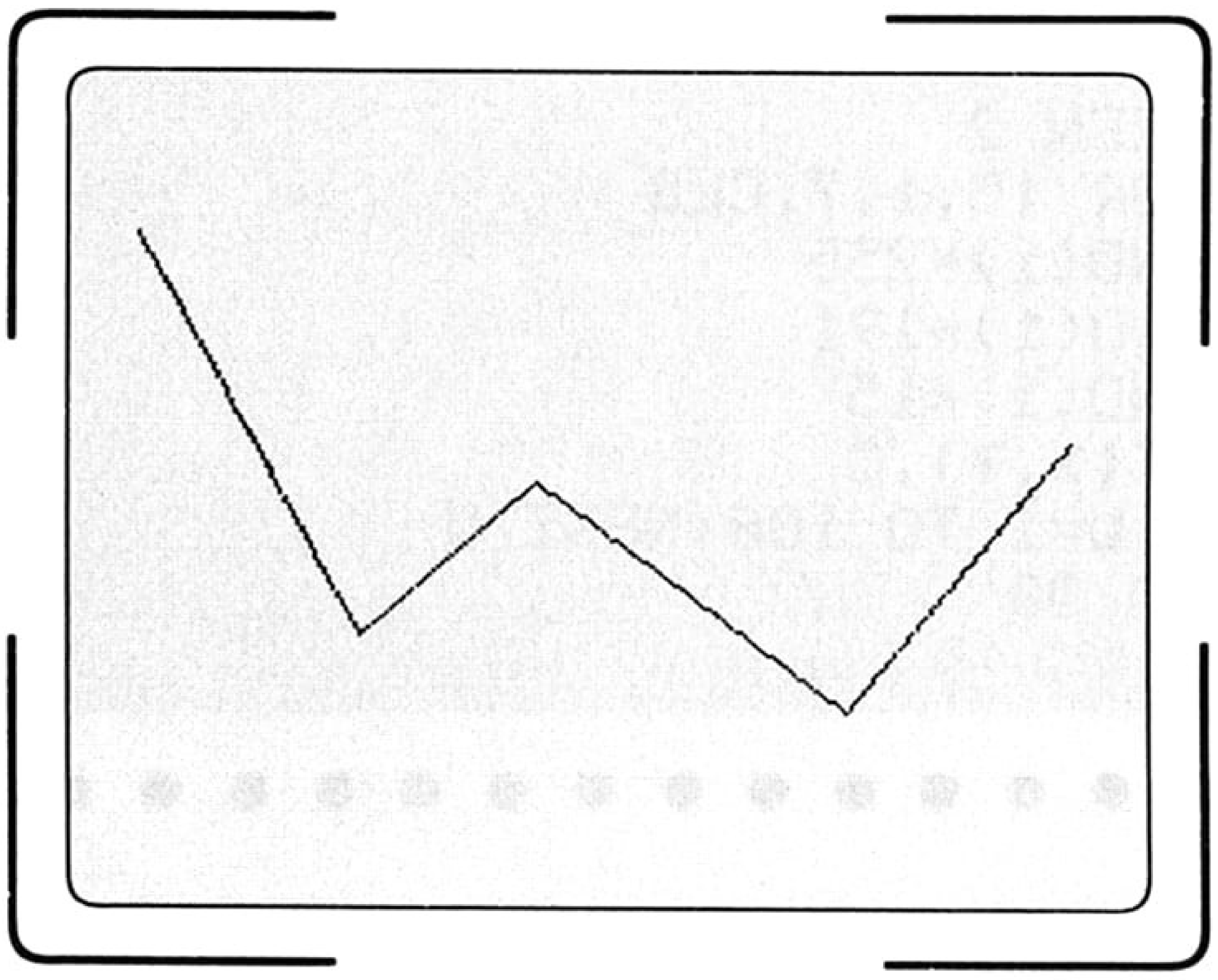
LINE命令は, すでに述べたように

LINE(引きはじめの位置座標)−(終わりの位置座標), 〔線の色〕

という形で使う. これで2点を結ぶ直線を望みの色で描くことができる. 直線のみでもけっこう美しい幾何学模様が描けるので, 例をあげておこう.

```
10 SCREEN 2
20 P1=3:P2=9
30 PSET(127,186),14
40 I=1
50 LINE-(SIN(I*P1)*100+127,COS(I*P2)*90+96),14
60 I=I+1
70 GOTO 50
```

さて，実際にLINE命令を使って図形などを描いていく場合には，ある地点まで直線を引き，次にそこから別の方向へ折れ曲がって進み，さらに別の方向へ進む，というふうに，折れ線を描いていくことも多い．



［折れ線を描く］

このように，続けてLINE命令を使う場合には，（はじめの位置）を省略すると，直前のLINE命令の（終わりの位置）が，自動的に（はじめの位置）とみなされる．これを利用すれば，プログラムをぐっと短くすることが可能だ．たとえば，上の図のような折れ線を描くプログラムは，次のように書ける．

```
10 SCREEN 2
20 COLOR 15,1,7:CLS
30 LINE(10,50)-(60,120),15
40 LINE-(100,80),15
50 LINE-(170,140),15
60 LINE-(220,70),15
70 GOTO 70
```

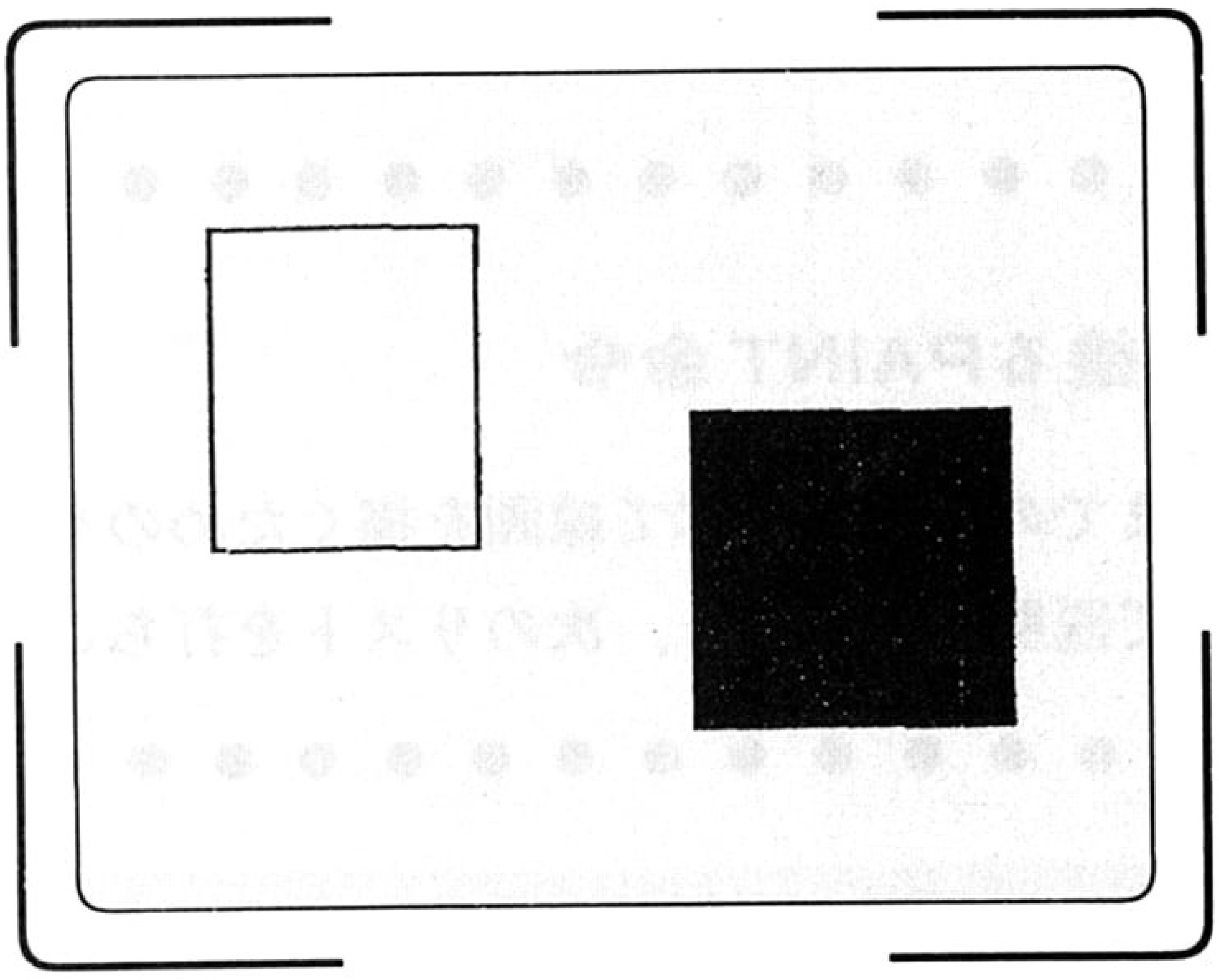
■LINE 命令とB, BFの指定

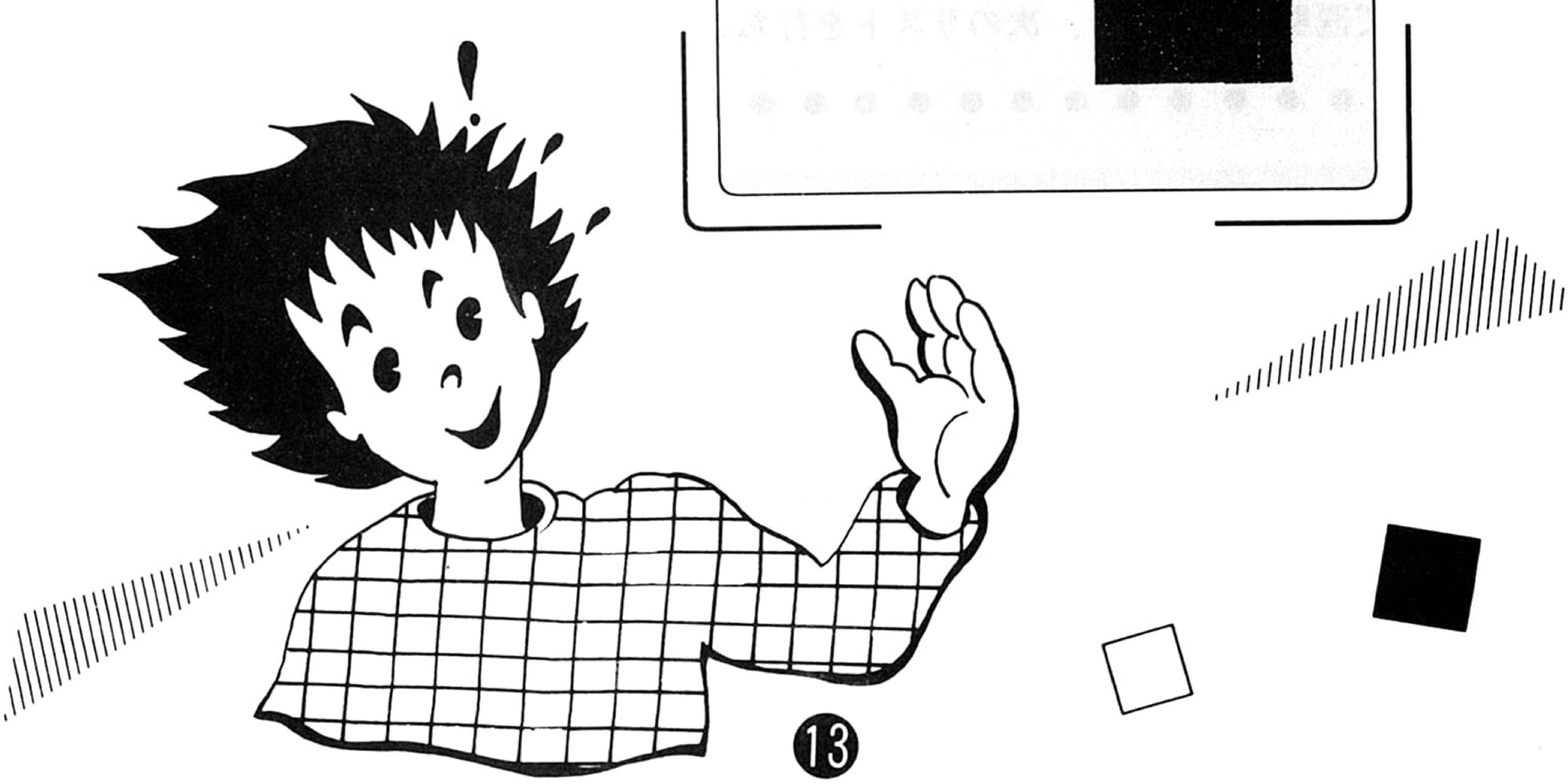
LINE 命令は本来の「直線を引く」という機能の他に,「長方形を描く」という機能ももっている. 次のリストを見てほしい.

```
10 SCREEN 2
20 COLOR 15,1,7:CLS
30 LINE(50,40)-(100,110),3,B
40 LINE(140,80)-(200,150),6,BF
50 GOTO 50
```

リストのうち, 30行と40行が長方形を描くための命令だ. LINE 命令のおしまいに「B」をつけてやると, 画面には, 本来 LINE 命令で引かれるはずだった直線をちょうど対角線とするような長方形が描かれる. また,「BF」をつけた場合には長方形が描かれたのち, その中味が塗りつぶされる.

BとBFの指定





■円を描くCIRCLE命令

円を描くために用意されているのがCIRCLE命令だ．一つの円を指定するためには，中心の座標と半径が決まればよい．CIRCLE命令は次の形式をとる．

CIRCLE(中心の座標),〔半径の長さ〕,〔円の色〕

これも例をあげておこう．実行とともに同心円がいくつも描かれるはずだ．なお，CIRCLE命令の場合には，画面の外に図形の一部がはみ出しても表示されないだけで，不都合はおこらない．

```
10 SCREEN 2
20 COLOR 15,1,7:CLS
30 FOR D=30 TO 150 STEP 20
40 CIRCLE(128,96),D,10
50 NEXT D
60 GOTO 60
```

■色を塗るPAINT命令

ここまでの命令はすべて線画を描くためのものだったが，最後に色を塗る命令について説明しておこう．次のリストを打ちこんで，実行してみてほしい．

```
10 SCREEN 2
20 COLOR 15,1,7:CLS
30 LINE(120,60)-(80,120),9
40 LINE-(160,120),9
50 LINE-(120,60),9
60 PAINT(120,90),9
70 GOTO 70
```

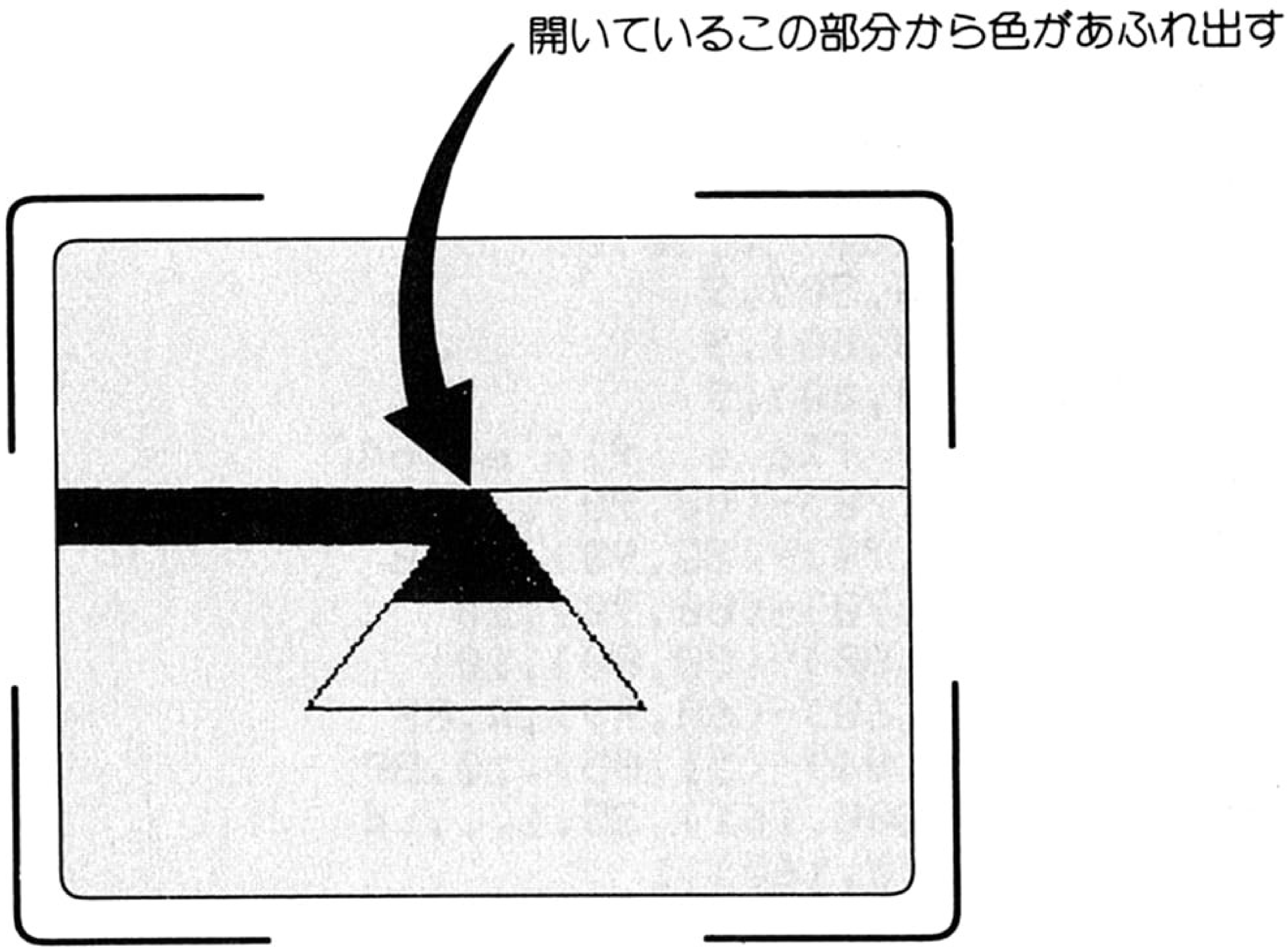
実行とともに，まず画面の中央に三角形が描かれ，続いてその内部が塗りつぶされたと思う．

PAINT 命令は，指定した色の線で囲まれた区域を塗りつぶす命令だ．

PAINT（塗りはじめの位置），〔塗りたい色〕，〔境界線の色〕

という形で使う．PAINT 命令を実行すると，まず（塗りはじめの位置）として指定した点に，あたかも絵の具を一滴落としたかのように「塗りたい色」が表示される．続いてその色は上下左右にひろがりはじめ，境界線として指定した色にぶつかるまでひろがり続ける．

このとき注意すべき点は，境界線が同じ色でつながっておらず，一部でも欠けて隙間があいていると，そこから塗りたい色があふれ出して，画面中が色の洪水になってしまうことだ．



【PAINTによる色もれ】

また，画面モード 2 の場合には，境界線の色は指定できず，つねに塗りたい色と同じと見なされるので，あらかじめ塗りたい区域の境界は，塗ろうとする色と同じ色で描いておく必要がある．これも覚えておこう．

1.3 MSXアニメ劇場

さて，これでMSX－BASICに用意されているグラフィック命令のうち，主なものについてひととおり目を通したことになる．最後にこれらの命令を組み合わせて，ちょっとした絵を描くプログラムを紹介しておこう．

次のリストは，伝統的な日本の冬の風景を描くプログラムだ．ただ静止画を表示するだけでは面白くないので，音を出すPLAY(プレイ)命令や，パターンを動かすSPRITE(スプライト)命令などを利用して，アニメ的（？）雰囲気を出してみた．実行とともに，「雪やこんこん」のメロディが流れ，犬や猫が画面に登場するはずだ．少し長いプログラムだが，ぜひとも打ちこんで試してほしい．

```
1000 COLOR 15,0,4
1010 SCREEN 2,3
1020 CLS
1030 PLAY "t120 l8 g. a16 g. a16 g4 e4"
1040 LINE(0,85)-(255,191),14,BF
1050 LINE(40,60)-(99,99),11,BF
1060 LINE(80,30)-(87,44),6,BF
1070 LINE(30,60)-(110,60),9
1080 LINE-(70,30),9
1090 LINE-(30,60),9
1100 PAINT(70,40),9
1110 PLAY "e. f16 e. f16 e4 c4"
1120 LINE(72,70)-(88,90),7,BF
1130 LINE(72,70)-(88,90),10,B
1140 LINE(80,70)-(80,90),10
1150 LINE(72,80)-(88,80),10
1160 LINE(48,68)-(60,99),4,BF
1170 LINE(50,84)-(51,85),10,BF
1180 CIRCLE(200,161),35,1,,,.4
1190 PAINT(200,160),1
1200 CIRCLE(180,130),40,15
1210 PAINT(180,130),15
1220 PLAY "l4 e c8. d16 e c8. d16"
1230 CIRCLE(180,80),30,15
1240 PAINT(170,80),15
1250 LINE(164,67)-(175,72),1,BF
1260 LINE(186,67)-(197,72),1,BF
1270 CIRCLE(170,80),4,1
```



```
1280 PAINT(170,80),1
1290 CIRCLE(192,79),4,1
1300 PAINT(192,79),1
1310 LINE(179,79)-(183,87),1,BF
1320 LINE(175,90)-(187,94),1,BF
1330 PLAY "e8. e16 g8. g16 d8. e16 d"
1340 FOR I=0 TO 3
1350 A$="":B$=""
1360 FOR J=1 TO 16
1370 READ S$
1380 A$=A$+CHR$(VAL("&H"+LEFT$(S$,2)))
1390 B$=B$+CHR$(VAL("&H"+RIGHT$(S$,2)))
1400 NEXT
1410 SPRITE$(I)=A$+B$
1420 NEXT
1430 DATA 0000,0000,0000,2000,7002,f804,3c08,7ff8
1440 DATA 1ff8,0ff0,0ff0,0c30,1a58,3248,0000,0000
1450 DATA 0000,0000,0000,0004,100e,101f,103f,1ffc
1460 DATA 1ff8,0ff0,0ff0,0c30,1668,1348,0000,0000
1470 DATA 0000,0000,0000,0000,0000,0000,3fff,1ffc
1480 DATA 1ffc,3ffe,3ffe,7fff,ffe7,0f80,0000,0000
1490 DATA 0000,0800,0980,1f00,1f80,0f80,1f80,1f00
1500 DATA 3f00,3f00,3f00,3f00,1f00,0000,0000,0000
1510 X1=0
1520 IF PLAY(0) THEN 1520
1530 PLAY "g8. a16 g8. a16 g8. g16 e8. e16"
1540 PLAY "e8. f16 e8. f16 e8. e16 c"
1550 X=RND(1)*100+10:Y=RND(1)*40+100
1560 IF X<X1 THEN PUT SPRITE 2,(X,Y),10,0:GOTO 1580
1570 PUT SPRITE 2,(X,Y),10,1
1580 X1=X:I=0
1590 IF I=55 THEN 1550
1600 I=I+1
1610 IF PLAY(0) THEN 1590
1620 PLAY "a g8. e16 l8 g g e c e e d d c4"
1630 FOR I=-16 TO 80
1640 PUT SPRITE 0,(I,120),9,2
1650 PUT SPRITE 1,(I-10,119),15,3
1660 NEXT
1670 GOTO 1670
```

TWO

Part 2

Let's get C.G.

パソコン お絵描き教室

―アニメ・キャラクタの描き方―

基本的なグラフィック命令の理解もできたところで，いよいよ本格的なアニメ・キャラクタを描く方法について考えてみよう．

パソコンで絵を描くためには，LINE命令やPAINT命令などを組み合わせていけばよい，ということはすでに述べた．アニメ・キャラクタのように，こみ入った絵の場合にも，基本的にはこのやり方で描くことができる．

けれども問題なのは，相手が「ユウちゃん」などのアニメ主人公ともなると，「雪ダルマ」のごとき「マルに目鼻」のものとはちがって，データの量が膨大になってしまうという点だ．BASICのグラフィック命令は，座標で位置を指定するという性質上，幾何学的な模様やグラフを描くのには向いていても，人の顔などの不規則な形を描くワザは，あまり得意じゃない．ちょっとした女の子の笑顔を描くにも，何十本，何百本ものラインを組み合わせる必要がでてくる．

そこで，絵の情報を座標データに移しかえるという，なんともやっかいな作業を，いかに手際よく，はやく，楽にすませるか，ということが重要になる．

データの入力を簡単にする方法はいくつか考えられるが，主なものとしては，

1．グラフ用紙を利用する．

2．透明なシートに絵を写しとり，ディスプレイの表面に貼りつける．そうしておいてから，お絵描き用のソフトウェア・ツールの助けを借りて，シート上の線をなぞる．

3．タブレットなど入力用の特別なハードウェアを用意する．

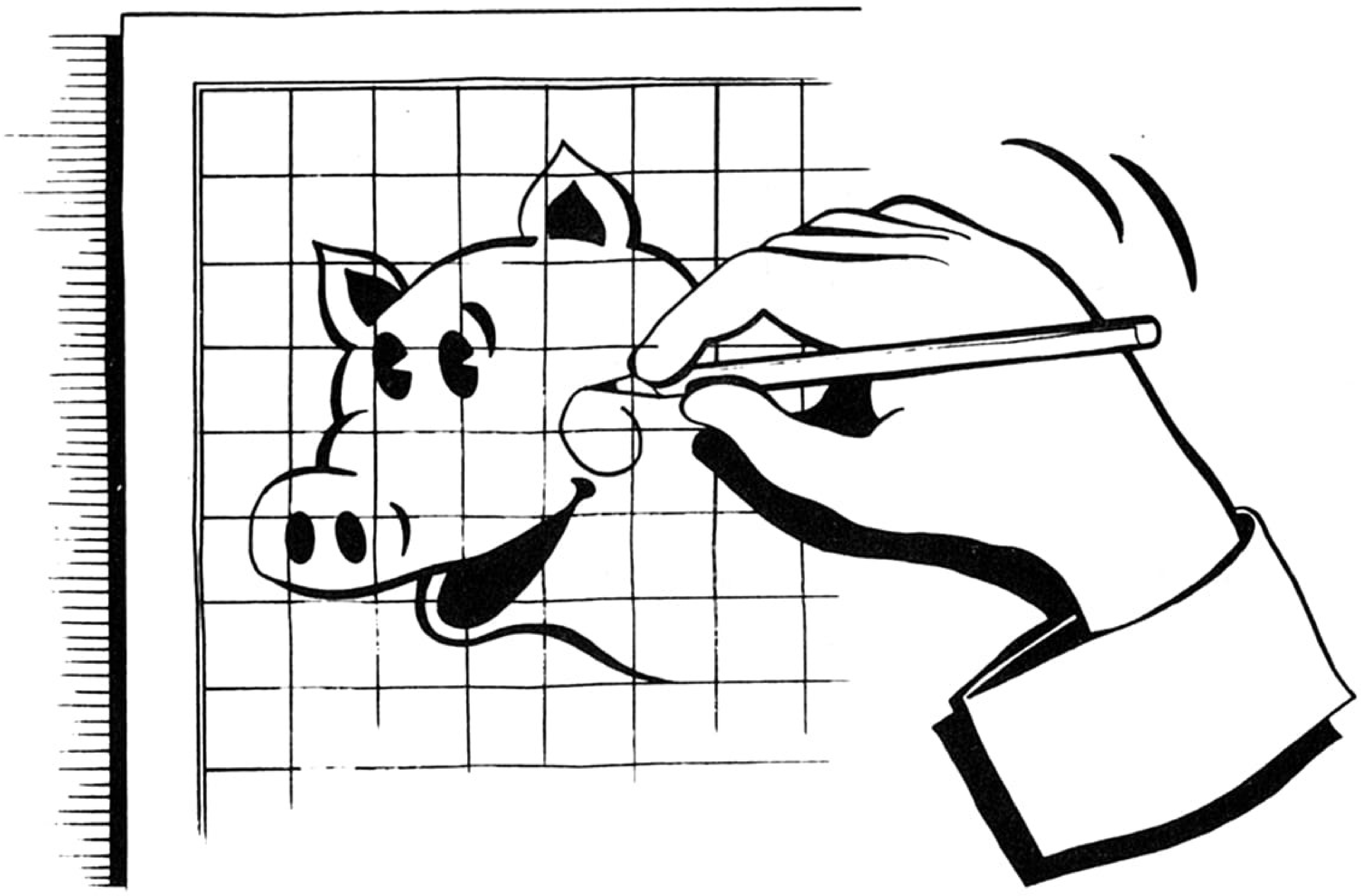
などがある．

それではこの3つの方法について，詳しく見ていこう．

2.1 グラフ用紙を利用する

グラフ用紙を利用する方法は，次のとおりだ．

まず，適当な大きさの1ミリ方眼のグラフ用紙を用意して，よこ256×たて192mmの，画面と同じ比率の枠を作る．次に，枠のなかに目的のアニメ・キャラクタの図柄を書きこむ．絵ができ上がったら，使われている一本一本の線の座標をグラフ用紙の目盛によって読みとり，パソコンに命令として入力していく．と，これだけだ．



アニメ・キャラクタをグラフ用紙に書いてみる

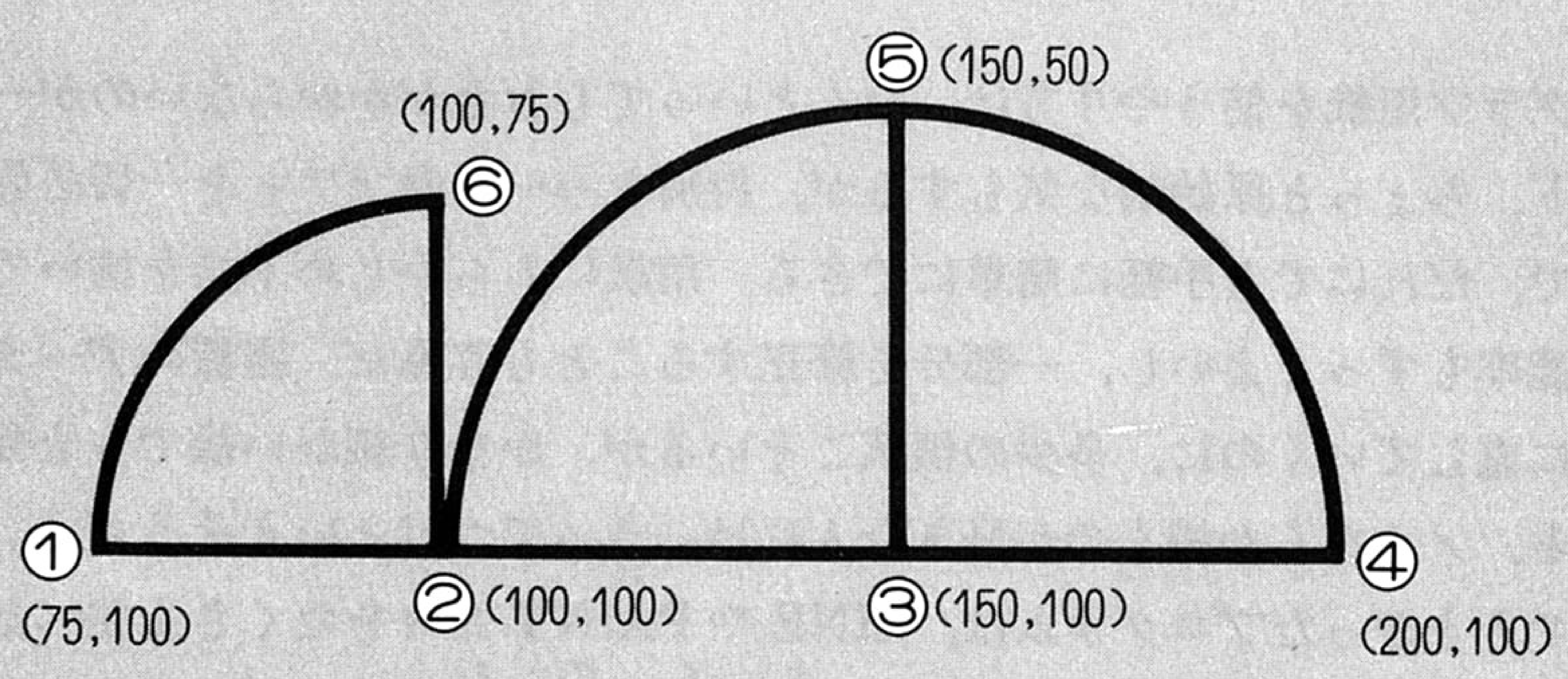
グラフ用紙を使うやり方は，なんといってもお金がかからないのが一番のメリット．ちょっと原始的な気もするが，特別なハードやソフトを一切必要としないので，だれにでも手軽に簡単にできる．用紙にあらかじめ下絵を描いてしまうので能率もずっとよいし，一部分を修正することも容易だ．座標のデータを命令の形に直していくのに，多少の根気こそいるが，かなり細かい絵でも十分に対応できる．メカなんか描くのが好きな人には，うってつけといえそうだ．

でき上がったプログラムは，LINE や PAINT 命令をたくさん並べたものとなるが，長すぎてわずらわしければ，READ(リード)～DATA(データ) 命令を使って省略をはかる手もある．例として，この形のプログラムを紹介しておこう．

```
100 SCREEN 2
110 COLOR 15,1,14:CLS
120 READ X,Y,C
130 IF C=-3 THEN 130
140 IF C=-2 THEN 180
150 IF C=-1 THEN 210
160 LINE-(X,Y),C
170 GOTO 120
180 READ X,Y,C
190 PAINT(X,Y),C
200 GOTO 120
210 READ X,Y,C
220 PSET(X,Y),C
230 GOTO 120
240 DATA 0,0,-1,120,60,9,80,120,9
250 DATA 160,120,9,120,60,9
260 DATA 0,0,-2,120,90,9
270 DATA 0,0,-1,100,40,3,70,100,3
280 DATA 130,140,3,100,40,3
290 DATA 0,0,-2,85,85,3
300 DATA 0,0,-3
```

ここで，グラフ用紙から線の位置を読みとるときに気をつけるべき事柄をあげておく．

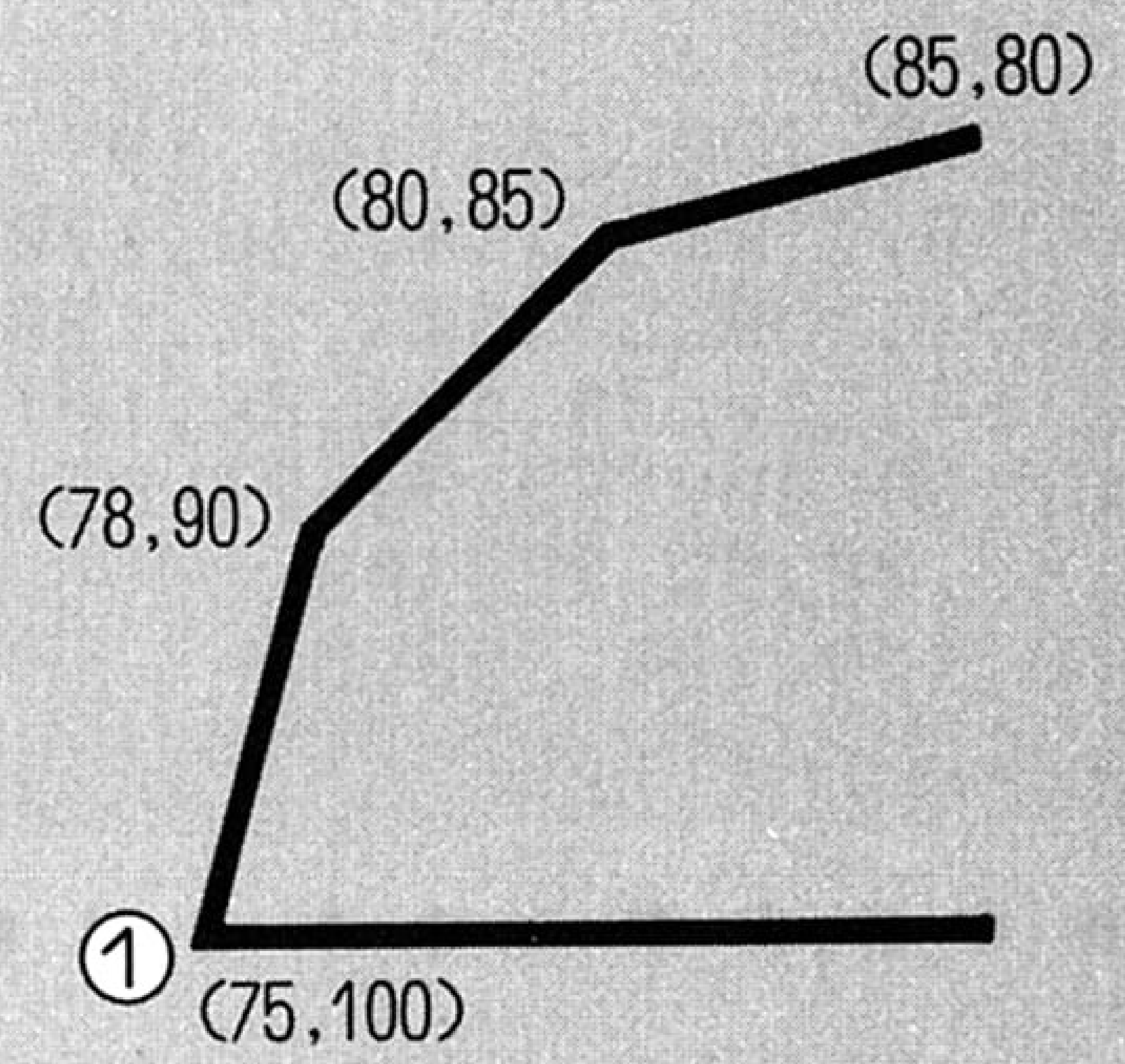
例として下の図のような形を描く場合を考えてみる．



グラフ用紙に描いた図形から座標を読む

まず，図の点①と点④のように，直線で結ばれている部分については，とくに問題はない．2つの座標をそのまま読みとって，LINE命令で結んでやればよい．

①と⑥，②と⑤のように，曲線になっているところは，短い直線に区切って少しずつつなげていくようにする．どのくらいの長さで区切っていくかは，画面の解像度(画面を構成する点の細かさ)と，曲線の曲がりぐあいによる．



曲線部分は細かく区切って直線のあつまりとして入力する

あまり荒くすると，曲線らしさが消えてしまうが，細かすぎてもかえって線がガタガタして見にくくなる．MSXの場合，解像度が256×192とわりに低いほうなので，図の①と⑥を結ぶ曲線ならば5 mm間隔ぐらい，②と⑤のときで1 cmぐらいに区切ればよい．できるだけだいたんなラインで簡潔に，という絵心が大切だ．

このほか，図の③や⑤のように，直線が集まる分岐点の座標はかならずデータとして入力しておこう．分岐点がぴたりと一致しないと，あとでPAINT命令で色を塗ったときに，色があふれだして悲惨な結末を招くからだ．

もうひとつ，線を描いていく順序についてもふれておこう．

たくさんの線があるときには，なるべく一筆描きで，描けるところは続けて描いてしまったほうがよい．直線が連続しているときには，始点の座標データは省略できるので，全体としてデータの数がぐっと少なくなる．図の例の場合には，①—④—⑤—②—⑥—①と結んでから，最後に③と⑤を結ぶようにすれば，少ないデータですむわけだ．

```
10 SCREEN 2
20 READ X,Y
30 IF X=-1 THEN 60
40 LINE-(X,Y)
50 GOTO 20
60 IF Y=-1 THEN 60
70 READ X,Y
80 PSET(X,Y)
90 GOTO 20
100 DATA -1,0,75,100,200,100,199,90
110 DATA 196,80,190,70,180,60,170,54
120 DATA 160,51,150,50,140,51,130,54
130 DATA 120,60,110,70,104,80,101,90
140 DATA 100,100,100,75,90,77,85,80
150 DATA 80,85,78,90,75,100
160 DATA -1,0,150,100,150,50
170 DATA -1,-1
```

以上あげた注意点は，ツールを利用して絵を描くときにもあてはまるので，心に留めておこう．

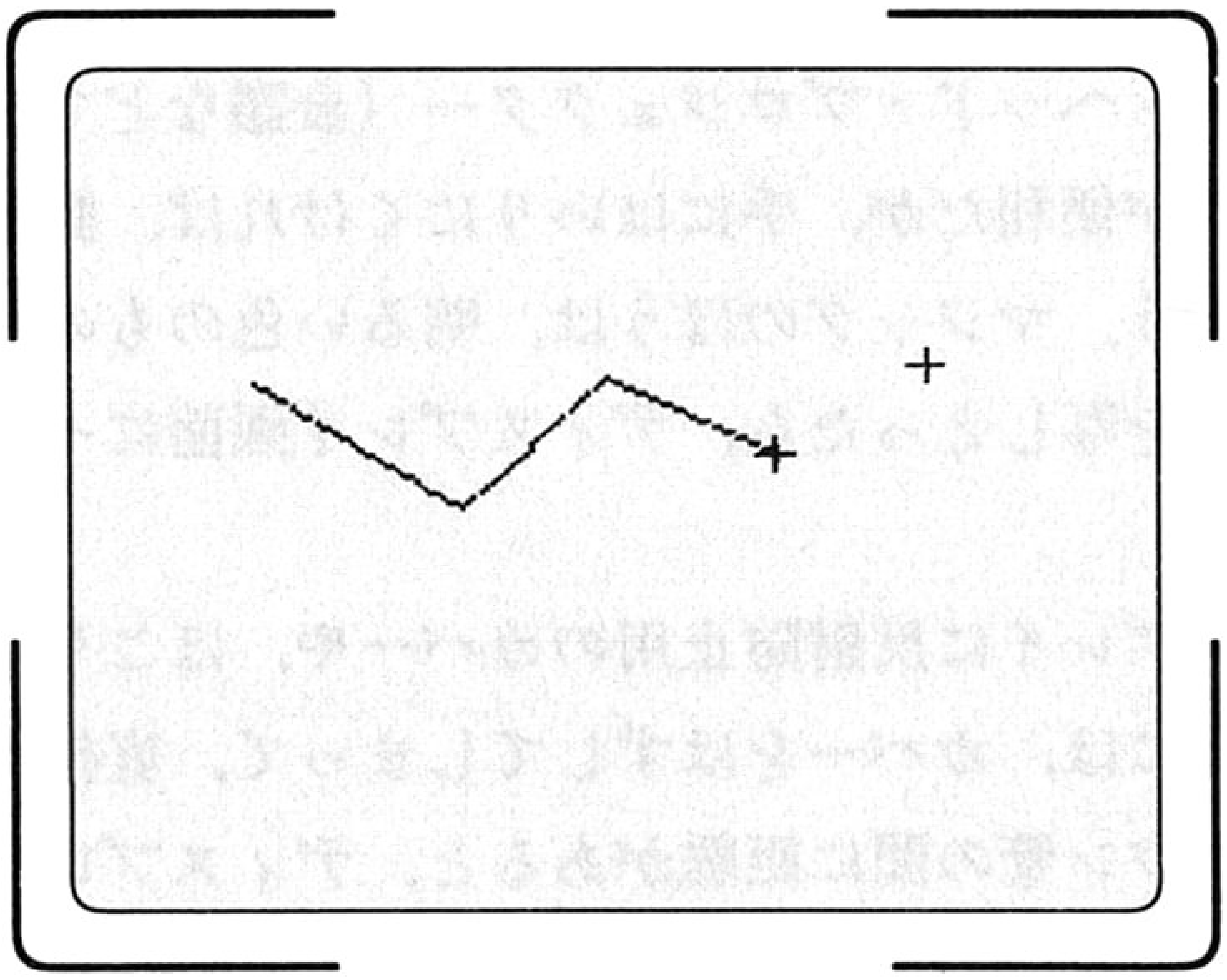
2.2 ソフトウェア・ツールを利用する

では，次にソフトウェア・ツールを利用して絵を描く方法に移る．

ソフトウェア・ツールというのは，一般にある作業をするための手助けとなるようなプログラムのことだ．たとえば，BASICのグラフィック命令を組み合わせて絵を描くときには，絵の情報のデータ化→入力→保存，とめんどうな手続きを山ほど踏まねばならない．そこで，この作業の大半を，パソコン自身にやらせてしまうプログラムを，あらかじめ作っておいて利用するわけだ．「お絵描き」用のツール・プログラムは，ふつう「グラフィック・ツール」とか，「グラフィック・エディタ」などと呼ばれる．

本書では，MSXパソコン用の「お絵描き」ツールとして，簡単なものから本格的・実用的なものまで，3つのプログラムを用意した．くわしい使い方などは3章にゆずるが，いずれも画面に表示されるカーソルを動かしながら，線を引いたり，色を塗ったりしていく仕掛けになっている．

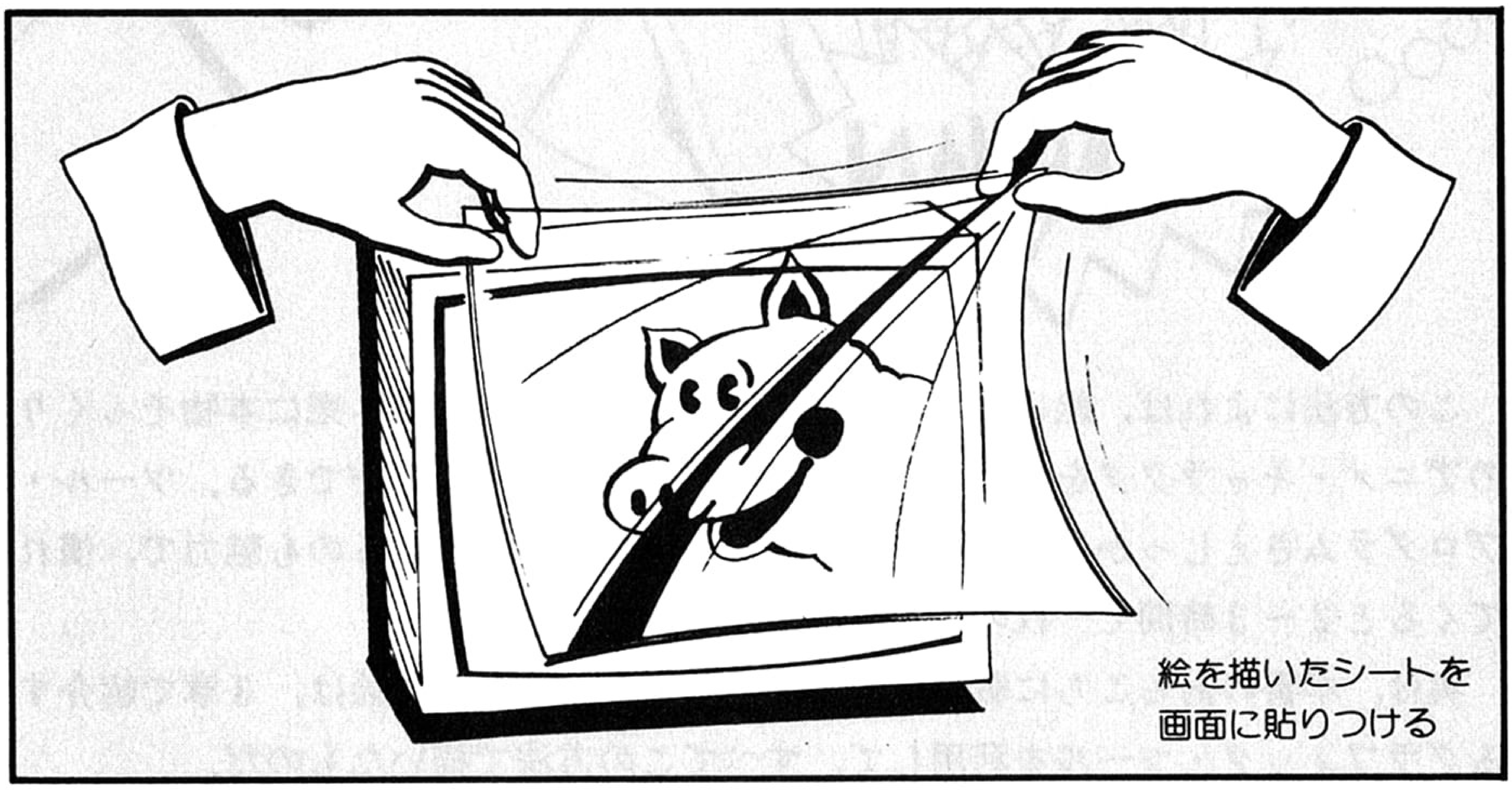
たとえば，一本の直線を引く場合には，カーソルを直線の始点の位置に合わせて，キーを押す．次に終点の位置までカーソルを移動させて，再びキーを押すと，2点を結ぶ直線が引かれる．



画面のカーソルを動かしながら線を引いていく

こうしたツールを使えば，あたかも紙の上で鉛筆を動かしていく感覚で，座標のことなどまったく気にかけずに絵を描けるわけだ．もちろん，描いていく手順は，すべてデータとして自動的にパソコンに記憶されていくので，あとで何度でも同じ絵を再現できる．

さて，ここで，こうしたツールを使う場合に，さらに作業を楽にするとっておきのワザを教えてしまおう．



絵を描いたシートを画面に貼りつける

まず，透明なシートを一枚と，油性のマジックインキを一本用意する．次に，シートに絵を描くのだが，腕に自身のある人は，直接オリジナルの絵をかきこんでもいいし，アニメ雑誌の気に入った絵の上にシートを重ねて，上からマジックでなぞってもいい．

シートはオーバー・ヘッド・プロジェクター（会議などで図表を映すのにつかうアレだ）用のものが便利だが，手にはいりにくければ，野菜などを包む台所用のラップ材で間に合う．マジックのほうは，明るい色のものを使おう．シートに絵の輪郭と，主な線を写しとったら，ディスプレイ画面にセロテープで貼りつける．

このとき，ディスプレイに反射防止用のカバーや，ほこりを防ぐためのガラスがかぶせてある場合には，カバーをはずしてしまって，直接ブラウン管の表面に貼る．シートとブラウン管の間に距離があると，ディスプレイを見る角度によって，線の位置がずれて見えるからだ．

貼りつけが終わったら，あとはシート上の線に沿って，カーソルを動かしながら絵を描くだけだ．

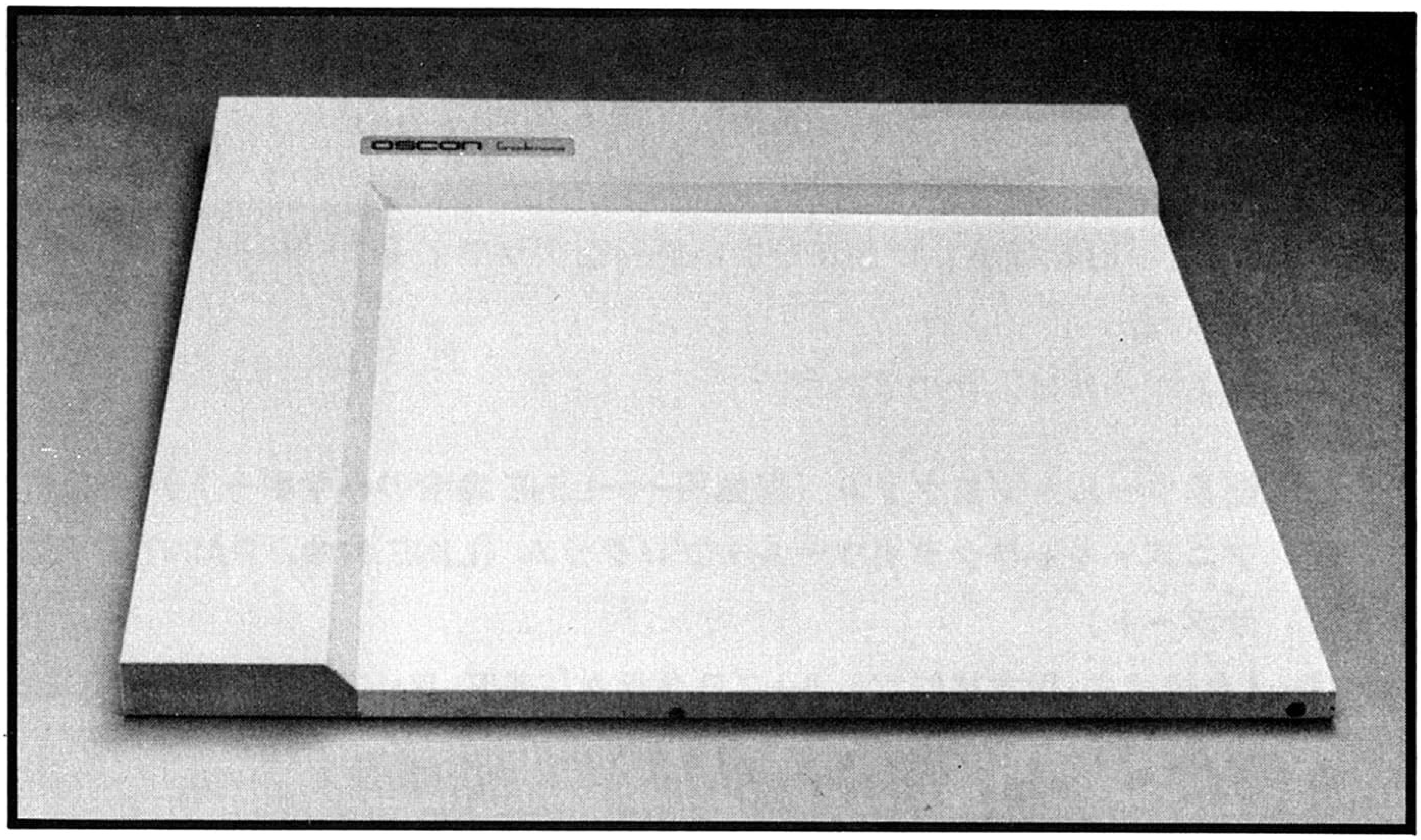


この方法によれば，絵にまったく覚えがない人でも大丈夫．楽に本物そっくりのアニメ・キャラクタを描いて，みんなに見せびらかすことができる．ツール・プログラムさえしっかりしていれば，とてもはやく完成できるのも魅力で，慣れてくると2～3時間で一枚の絵が描ける．

実は，本書のあちこちに載せたアニメ・キャラクタたちの絵は，3章で紹介するグラフィック・ツールを利用して，すべてこの方法で描いたものだ．

2.3 入力用のハードウェアを利用する

最後に，ちょっとお金はかかるが，たいへんに効率がよい方法として，タブレットなど特別な入力装置を利用する方法を紹介しておこう．



タブレット

タブレットというのは，写真のように製図板のような形をしている．この板の上で，照準を動かしながらスイッチを押していくと，そのまま照準の位置を座標データとしてパソコンに伝えてくれる．紙に描いた原画をタブレットの上にのせて，照準でなぞっていけば，たちどころに絵のデータがとりこまれるという，たいへんに便利な装置だ．

MSXの場合には，BASICにもタブレット関係の命令が用意してあり，将来的な拡張の余地もある．近いうちに安価なタブレットが登場すれば，「お絵描き」用として十分実用できるだろう．アニメ派，グラフィック派のみんなにとっては，必需品となる日も来るかもしれない．

このほか，ライトペンなどを備えたMSX機種の場合は，これを利用してより使いやすいグラフィック・ツールをあつらえることも考えられる．

THREE **3** Part

Let's get C.G.

MSX「お絵描き」用ツール

この章では，「お絵描き」にかかせない便利なグラフィック・ツールのプログラムを紹介しよう．

とり上げるツールは，次の3つだ．

① 簡易ツール・プログラム（線画用――LINE命令のみサポート）

② アニメ・キャラクタ用ツール・プログラム（LINE命令，PAINT命令をサポート）

③「お絵描き」用本格的ツール・プログラム(LINE，B・BF，PAINT，CIRCLEなど，すべてのグラフィック命令をサポート）

いずれも画面に表示されるカーソルを動かして，線を引いたり，色を塗ったりする仕組みになっている．①から順に機能的にアップしているので，用途に合わせて利用しよう．





3.1 プログラムを入力する前に

各ツールの説明にはいるまえに，プログラムの入力方法から説明しておこう．それぞれのツール・プログラムを，一字一句たがわぬように，MSX に入力していく．プログラムは，1字のまちがいでも動かなくなることがあるので，確実に，確かめながら進んだほうが，かえって時間の節約になるはずだ．

「うわーっ，なんて長いプログラムだ」なんて，ため息をつく人もいるかもしれないが，尻込みせずに，あせらず，根気よく入力してほしい．便利なソフトウェアがひとつ増えることは，きみのパソコンの能力を何倍も引き出すことなのだから．

この3つのツール・プログラムは，わずかだが機械語の部分を含んでいるので，ひととおり入力が終わったら，かならずテープにセーブしておくようにしよう．機械語は BASIC とちがって入力ミスがあったときには，暴走して全部のプログラムをぶち壊してしまうことがある．いったんセーブをしておけば，保険をかけたのと同じで，何度でもロードし直してまちがいをチェックすることができる．

セーブとベリファイ（プログラムがまちがいなくセーブされているかチェックすること．マニュアルの CLOAD ?の項を参照）が終わったら，さっそく RUN させてみよう．以下，実行中に「Syntax Error」などのエラーメッセージがでたら，命令語のつづりなどを確かめてみる．また，誤動作が起きた場合には，変数名や飛び越し先の行番号がまちがっていないか調べよう．それでは，ツールの使い方に移る．

3.2 簡易「お絵描き」ツール

はじめに紹介するのは，線画を描くための簡易「お絵描き」ツールだ．このプログラムでは，色塗りができないので，実用としてはちょっともの足りないが，グラフィック・ツールがどのようなものであるかを見ていただくために，ここにとり上げた．

39ページのプログラム「GT－1」を入力後，RUN させると，まず画面に，「ジョイスティックを使うか，カーソルキーを使うか」という質問が出るので，「j」か「k」のキーを押して答える．すると，画面がいったんクリアされて，中央に

白い十字型のカーソルが表示される．これがグラフィック用のキャンバスだ．

以後の操作は，すべてカーソルキー(またはジョイスティック)，ファンクションキー，英字キーを使って行う仕組みになっている．

■キーの使い方

以下，各キーの使い方を順に説明していこう．なお，英字キーの入力は，小文字でしか受けつけられないので，[CAPS LOCK]のロックと[かな]のロックははずしておく．

① [↑][↓][←][→] **カーソルキー(またはジョイスティック)**

画面上のグラフィック・カーソルを移動させる．

② [L] **キー**

直線の始点を指定する．望みの位置へカーソルを動かして，[L]キーを押すと，始点の位置がマークされる．

③ **スペースバー**（またはジョイスティックについているトリガーボタン）

直線の終点を指定し，同時に直線を描く．前の直線に続けて引きたいときには，カーソル移動→スペースバー→カーソル移動→スペースバーと押していけば，連続して線が描かれる．

④ [C] **キー**

直線の色の指定．１回押すごとに，カーソルの色と画面の外枠の色が変わる．以後は，外枠の色と同じ色で線が引かれる．

⑤ [B] **キー**

バックカラーの指定．１回押すごとに画面の背景の色が変わる．

⑥ [CLS HOME] **キー**

これまでに描いた絵をもう一度はじめから描き直す．

⑦ DEL キー

すぐ前に描いた直線を消す．もちろんデータのほうも直線一本分が削除される．

⑧ F・1 キー（ファンクションキーの１番）

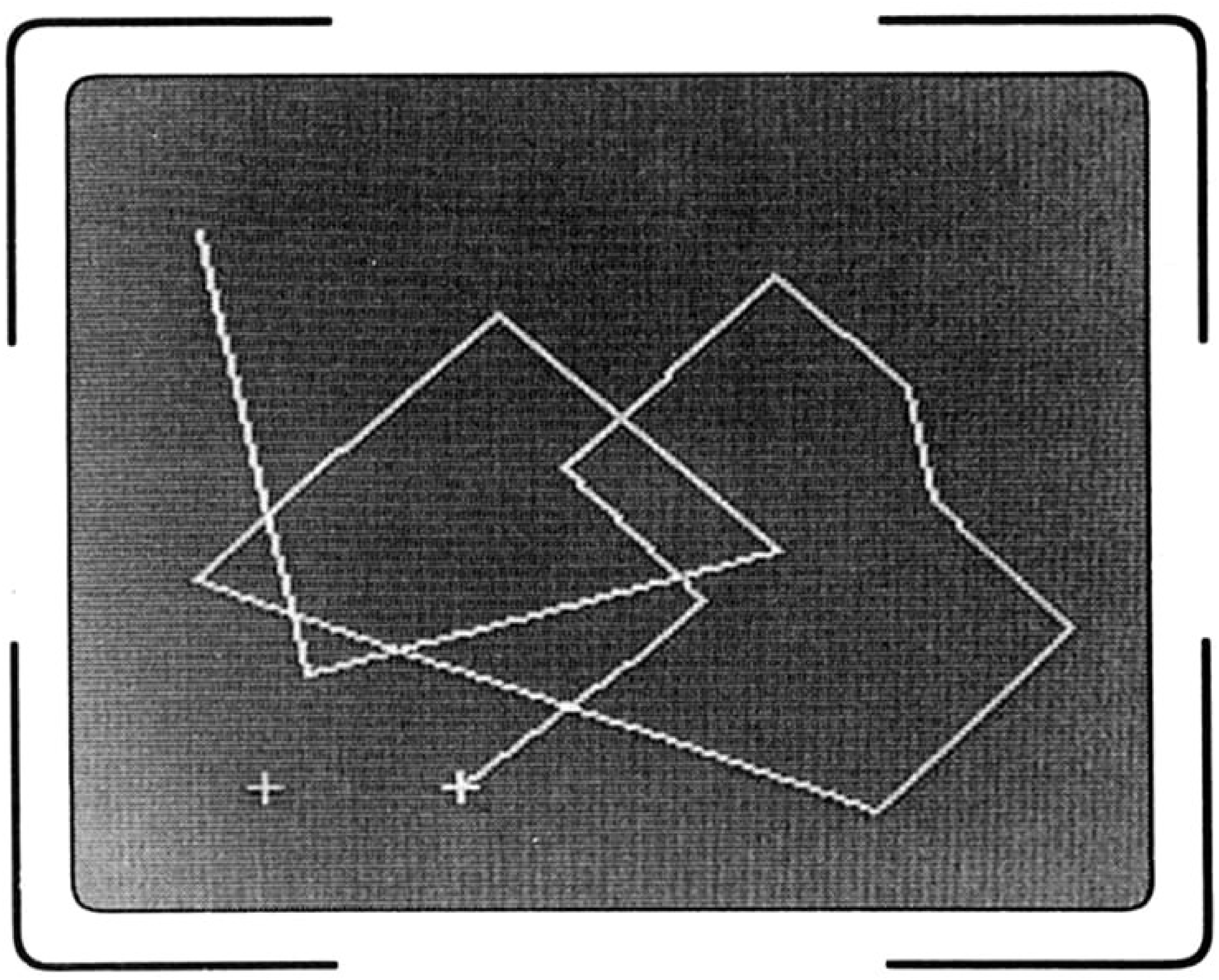
絵のデータをロードする．「File Name ?」の質問に答えてデータの名前（ファイル名）を入力したのち，リターンキーを押すと，次の F・2 の機能によってセーブしておいたデータがカセットから読みこまれる．

⑨ F・2 キー（ファンクションキーの２番）

データのセーブ．絵を描き終わったら， F・2 のキーを押すと，「File Name ?」の質問がでる．適当なファイル名を入れて，リターンキーを押すと，絵を描く手順がデータとしてカセットに送りこまれる．ここでセーブしたデータは，このプログラムの F・1 の機能を使わないとロードできない．描きかけの絵をとっておくときに便利．

⑩ F・3 キー（ファンクションキーの３番）

BASIC プログラムの生成．このキーを押すと，なんと，たった今描いたばかりの絵とまったく同じものを描き上げる BASIC のプログラム（ここがミソ！）が作り出される．「BASIC FILE NAME ?」の問いに対してファイル名をつけてやると，プログラムが作られて，カセットにセーブされる．なお，このプログラムはセーブ形式が異なるので，ロードするときには，「LOAD "CAS：ファイル名"」（CLOAD ではない）と命令するように．

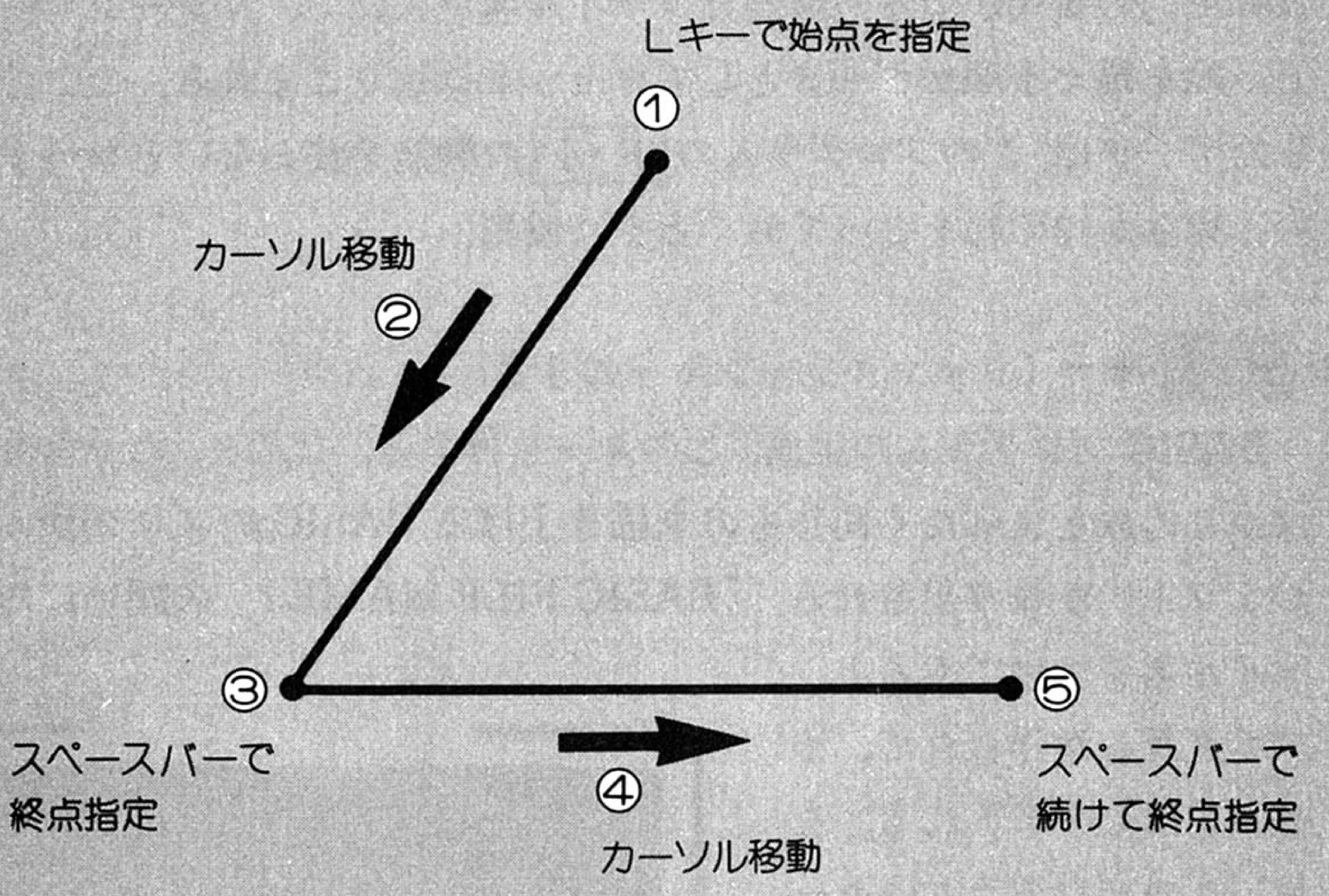


■三角形を描いてみる

では，このツールで何か描いてみることにしよう．とりあえず，三角形をひとつ描いてみる．

まず，背景の色を決める．何回か B キーを押して，好きな色を選ぼう．

次に L キーを押すと，カーソルが現在の位置にマークされる．ここを頂点として，カーソルを左斜め下に動かす．適当なところで，スペースバーを押すと，2点が直線で結ばれたはずだ．さらにカーソルを真横に持っていって，再びスペースバーを押す．これで三角形の2辺が描かれた．3本目の場合も同じように，はじめの地点までカーソルをもどして，スペースバーを押せばよい．



[三角形を描く手順]

どうだろうか．ツール・プログラムを利用すれば，このように自由自在に絵が描けるわけだ．

簡単な絵を描いたら，セーブやロードなど，他のキーの機能も試してみよう．なおセーブのときには，カセットを録音状態にしておくことをお忘れなく．また，ファイル名は6文字までしか使えない．念のため．

3.3 アニメ・キャラクタ用ツール

次にアニメ・キャラクタを描くための，本格的なグラフィック・ツールの説明に移る．前にも述べたが，本書におさめた「ユウちゃん」などの絵は，すべてこのツールを使って描いたものだ．

このツールには，線画を描くLINE機能に加えて，色を塗る機能，中間色表示の機能などが補強されている．MSXパソコンには，基本16色が用意されているが，これだけでは肌色などの微妙な色合いが出せず，せっかくのアニメ・キャラクタも雰囲気が出ない．そこで，2つの色を混ぜ合わせて，中間色が使えるようにした．もちろんデータのセーブ，ロード，BASICプログラムの生成なども，簡易ツールとまったく同じように行える．

44ページ以下のプログラム「GT−2」を，まちがえないように入力する．リストの320行から410行まで，さらに9440行〜9470行までは，機械語のデータなので特に注意してほしい．くどいようだが，入力が終わったら，実行するまえにかならず一度テープにセーブしておくことを忘れないようにしよう．

■キーの使い方

では，各キーの働きを説明する．線を引く，データをロード，セーブするなど，主な操作は3.2.の簡易ツールとほとんど同じなので，追加・変更となった機能についてだけ述べておこう．

①カーソルキー（またはジョイスティック）
②[L]キー
③スペースバー（またはトリガースイッチ）
④[C]キー

（①〜④：簡易ツールに同じ）

⑤ [B] キー

背景色の指定を行う機能は変わらない．ただし，[B]キーを押すと，「BACK COLOR = ?」と聞いてくるので，直接カラーコード（0−15）で指定する．

⑥ CLS HOME キー
⑦ DEL キー
⑧ F・1 キー
⑨ F・2 キー
⑩ F・3 キー

簡易ツールに同じ

⑪ P キー

ペイント（色塗り）機能．同じ色の境界線で囲まれた部分を，カーソルの位置から塗りつぶす．

⑫ T キー

タイリング（中間色で塗り変える）機能．あらかじめ基本色で塗っておいたところを，2つの色を混ぜ合わせた中間色で置き換える．手順は少し複雑だが，次のとおりだ．

まず，T キーを押すと，中間色置き換えモードにはいった印として，カーソルの点滅が少しはやくなる．そこで，カーソル移動→スペースバーと操作して，置き換えを行う範囲の上端と下端を指定する（横方向の指定はできないので，指定した上下範囲のうち画面の外枠の色と同じ部分がすべて置き換わる．

次にどの色とどの色を混ぜ合わせるかという質問が，「COLOR1＝？」および「COLOR2＝？」という順で出るので，カラーコードで答える．このとき画面の右上に，中間色のサンプルが表示され，「OK ？ (y/n)」（思いどおりの色かどうか）と確かめてくる．

Y を押すと，置き換えが実行され，N をおすと，再び色の指定をやり直せる．途中でやめたいときは，色指定のときにリターンキーを押す．

⑬ ? キー

このキーを押すと次のような情報を表示する．

・カーソルの現在位置の座標と，その点のカラーコード
・直前の命令の座標
・現在の線の色と背景の色（カラーコード）
・すでに使ったステップ数と残りのステップ数(LINE や PAINT などの手順

をあといくつ使えるか)

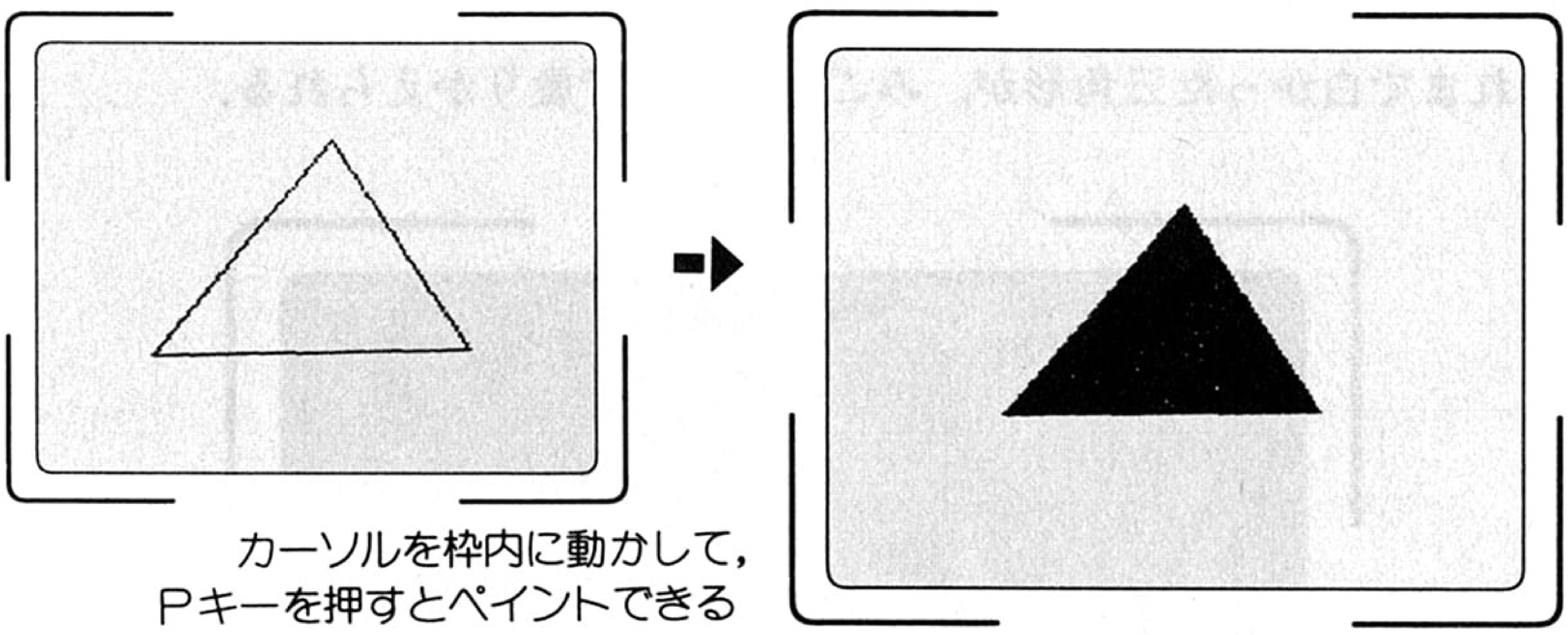
これらの情報は，[?]を押したあと，スペースバーを押していくと，次々に表示される．もとの画面にもどりたいときは，リターンキーを押せばよい．

■新しい機能(ペイントと中間色)

新しくつけ加わった⑪から⑬のキーについては，実際に試してみたほうがわかりやすいと思うので，例をあげて説明しよう．

まず，簡易ツールのときと同じように，カーソルを移動させながら，[L]キーとスペースバーを使って三角形をひとつ描いてみる．

描き終わったら，今度はカーソルを三角の中（どこでもよい）に動かして，[P]キーを押す．すると，三角形の中が辺と同じ色できれいに塗りつぶされたはずだ．



カーソルを枠内に動かして，
Pキーを押すとペイントできる

もしかしたら，三角形の外まで色があふれ出して，画面が色の洪水になってしまった人もいるかもしれない．ペイントをかける（色を塗る）ときには，境界線として囲った枠がぴたりと閉じていなくて，少しでも隙間が開いていると，そこから色があふれ出してしまう．

でも，ご心配なく．色があふれてしまったら，あわてずに[DEL]キーを押そう．ペイントをかける前の状態に，画面が復元される．

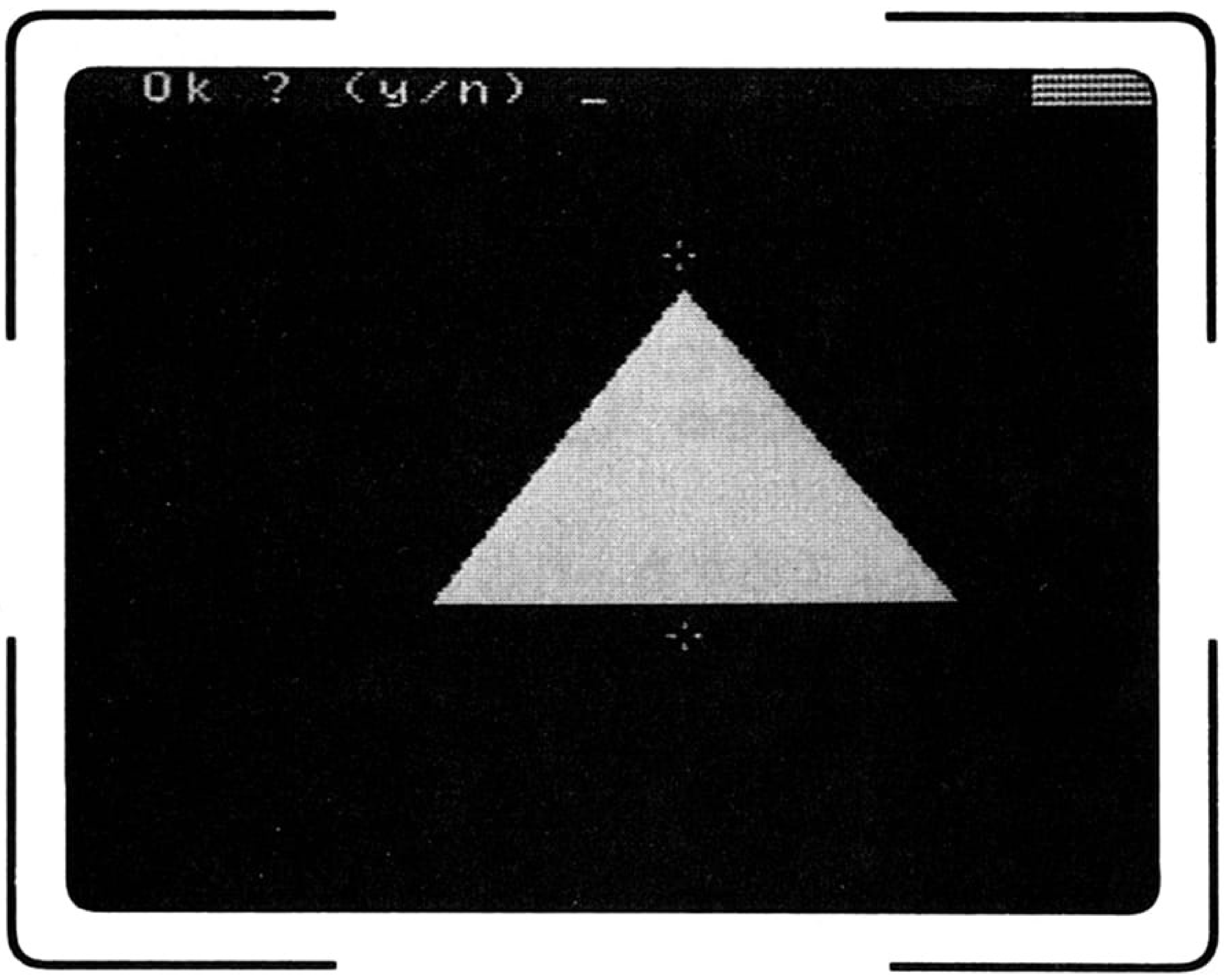
こんな色漏れ事故を防ぐためには，スペースバーで線を引いてしまう前に，ちゃんとカーソルが目的の位置に合っているかどうかを，[?]キーを押して確認しておこう．ついでに言っておくと，色塗りは境界線と同じ色でしかできないので，境界を囲ったあと，ちがう色で塗ろうとすると，画面全部が塗りつぶされてしまう．

このほかにも ? キーには記憶できる残りの手順の数を表示する機能がある．線引きや色塗りの手順は，全部で1,500ステップまでパソコンに記憶させられるが，1,500を越えてしまうとエラーと「Out of Memory」の表示が出てそれ以上のデータが入力できない．細かい絵を描くときには気をつけよう．

さて，もうひとつ，中間色の表示機能についても試してみよう．いま描いたばかりの白い三角形の中を，肌色に置き換える．

T キーを押したら，カーソルを三角形の頂点あたりに移動させて，スペースバーでマークする．カーソルは，8ドット単位で飛び飛びにしか動かないので，ぴったりあわせる必要はない．上端を決めたらカーソルを下辺近くに動かして，同じようにマークしてやる．

混合する2つの色には，「9＝明るい赤」と「15＝白」を選ぼう．画面の右上に，肌色らしきサンプルがあらわれたはずだ．ここでyキーを押すと，あら不思議，これまで白かった三角形が，みごとに肌色で塗りかえられる．



ところで，この中間色なるもの，よく見ると2つの色の線を交互に積み重ねた縞模様であることに気がつく．えっ，なになに，よく見なくても縞模様にしか見えないって？

そういう人は，ディスプレイから50cm以上はなれて見てみよう．きっと2つの色が入り混じって立派な肌色に見えるはずだ．

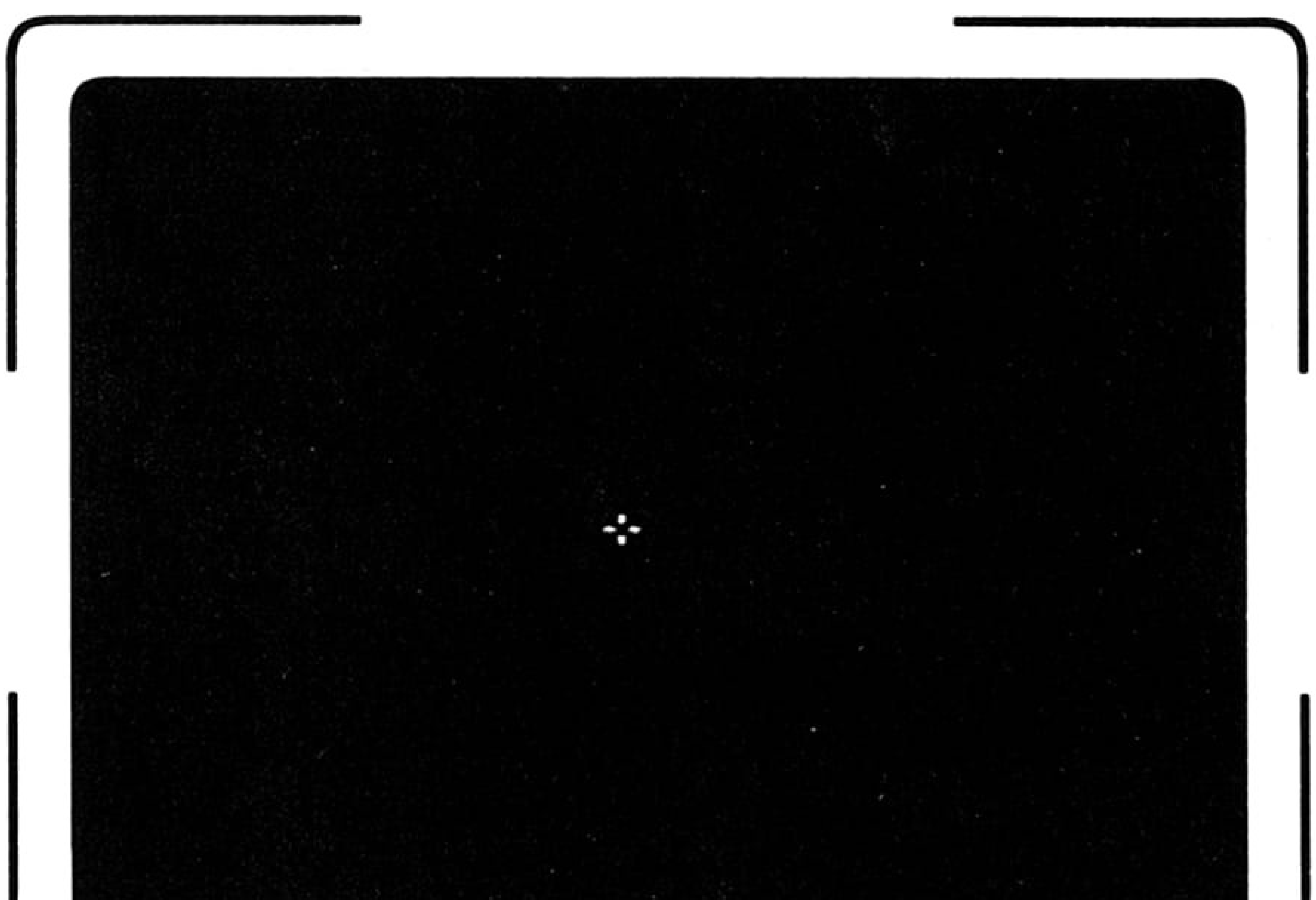
参考までに全部の中間色を表示するプログラムを紹介しておこう．

```
10 SCREEN 2
20 FOR I=0 TO 15
30  LINE(I*17,0)-(I*17+16,191),I,BF
40 NEXT
50 FOR I=1 TO 15
60  FOR J=1 TO 16 STEP 2
70   LINE(16,I*12+J)-(255,I*12+J),I
80  NEXT J
90 NEXT I
100 GOTO 100
```

以上でアニメ・キャラクタ用ツールの説明は終わり．13種のコマンドを駆使して，素敵なアニメ・キャラクタたちを完成させてほしい．

3.4 「お絵描き」ツール本格版

さて，３本目に紹介するのは，画面いっぱいに思いのままに落書きしていただくための，本格版「お絵描き」ツールだ．例によって，53ページからのリスト，「GT−3」を入力したら，カセットにセーブしたのち，実行してみる．はじめに，次の画面があらわれるはずだ．



この「お絵描き」ツールは，すべての操作をカーソルキーとスペースバー，ファンクションキーのみで行える．もちろん，カーソルキー，スペースバーの機能は，ジョイスティックに切り換えることもできる．各キーの役割を説明しよう．

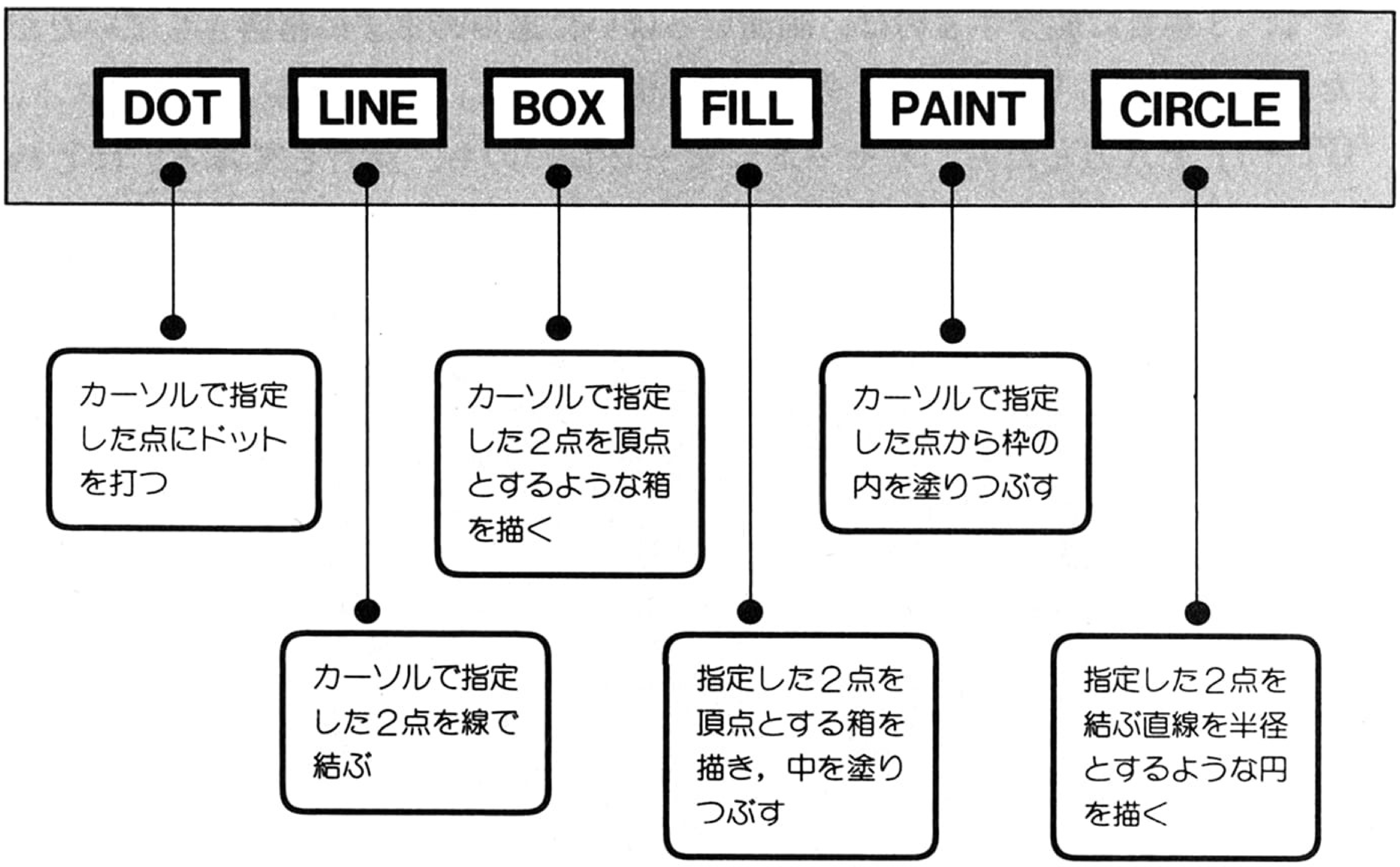
①カーソルキー，スペースバー

カーソルの移動と，座標の指定．このほか，画面に表示されるメニューから命令などを選択するのにも使う．

②ファンクションキー

F・1

LINE, PAINT などの操作を決める．F・1 キーを押すと，下の図のように命令のメニューがでるので，カーソルキーとスペースバーでこの中から選ぶ．



。カーソルの移動は↑↓←→のキーで行ない，点の指定はスペースバーで行なう．

F・2

画面データのロード.

F・3

カーソルの色を指定する. F・3 を押すと, 画面に色見本が出るので, その中から選ぶ.

F・4

線の太さを指定する. 太さは全部で3段階. F・4 を押すたびに変わる.

F・5

カーソルの移動のはやさを切り換える. SLOW/FAST の2段階. カーソルの位置を正確に決めたいときには, 遅いほうのSLOWモードに切り換えればよい.

F・6

カーソルキーと, ジョイスティックを切り換える. プログラム・スタート時には, カーソルキー・モードになっている.

F・7

画面データのセーブ.

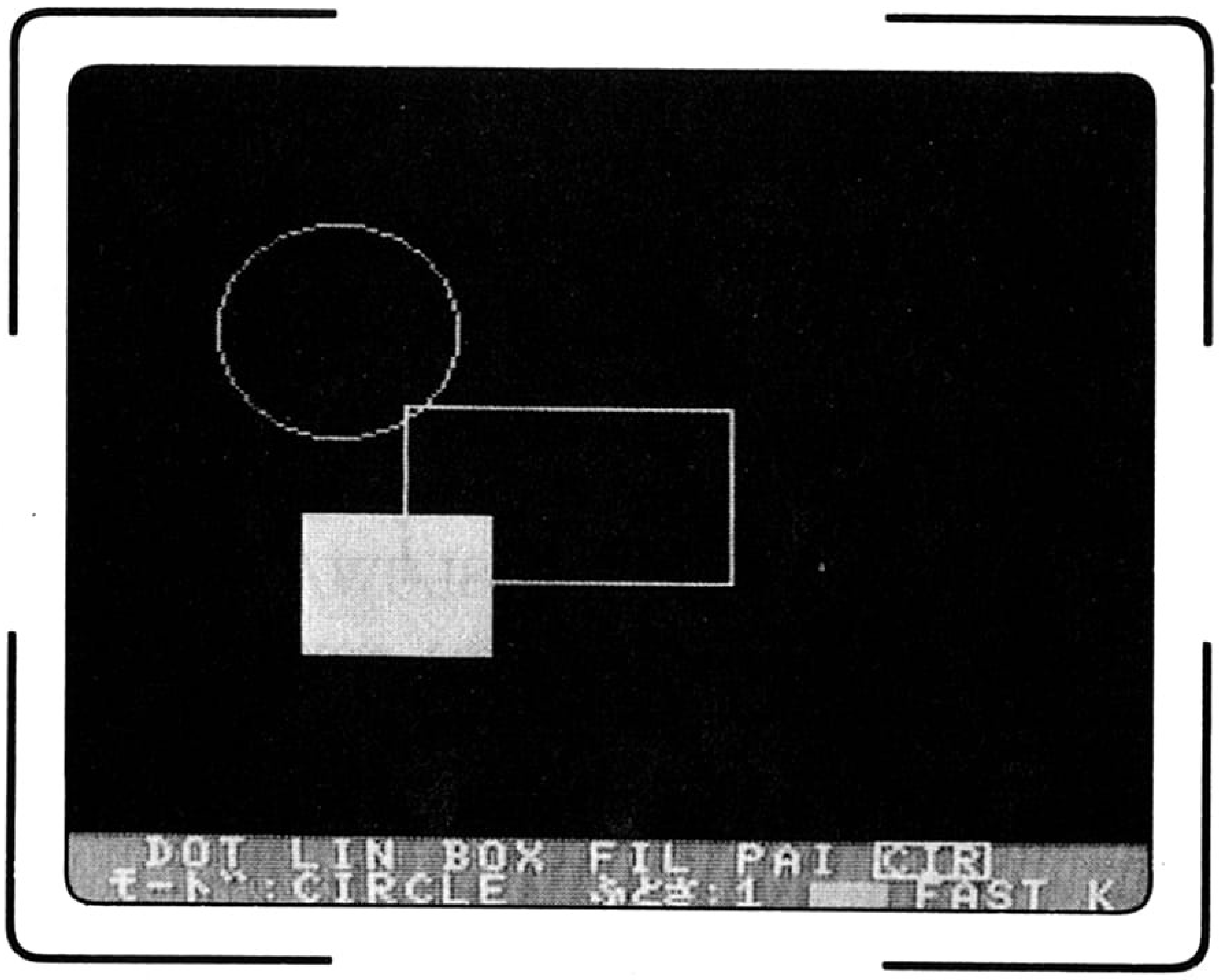
F・8

背景色を指定する. これも色見本から選ぶ.

F・9

画面に描いた絵を全部消去する. いわば黒板消し.

このグラフィック・ツールは，長方形や円を簡単な操作で描けるので，うまく組み合わせていけば，楽しい絵ができ上がるだろう．線の太さを変えられるのも，絵にアクセントをつける上でおおいに役立つ．下の図はこのツールを利用して描いた絵だ．



ひとつだけ注意しておきたいのは，PAINT 命令を使ったときにこのツールでは，いったん色があふれてしまうと，もとにもどすことができないということだ．PAINT は慎重に使おう．

また，まちがって F・2 や F・7 (ロードとセーブ) を押してしまったときには，ESC キーでもとにもどることができる．

さて，こうしてツールを使っていろいろな絵を描いてみると，MSX では，線の引き方によっては，まわりの色が影響を受けて変化してしまう場合が，たびたびあることに気づく．これは，画面モードが 2 のときには，色の使い方に制限があるからだ．次の章ではこの点について詳しく述べることにしよう．

簡易「お絵描き」ツール　GT-1

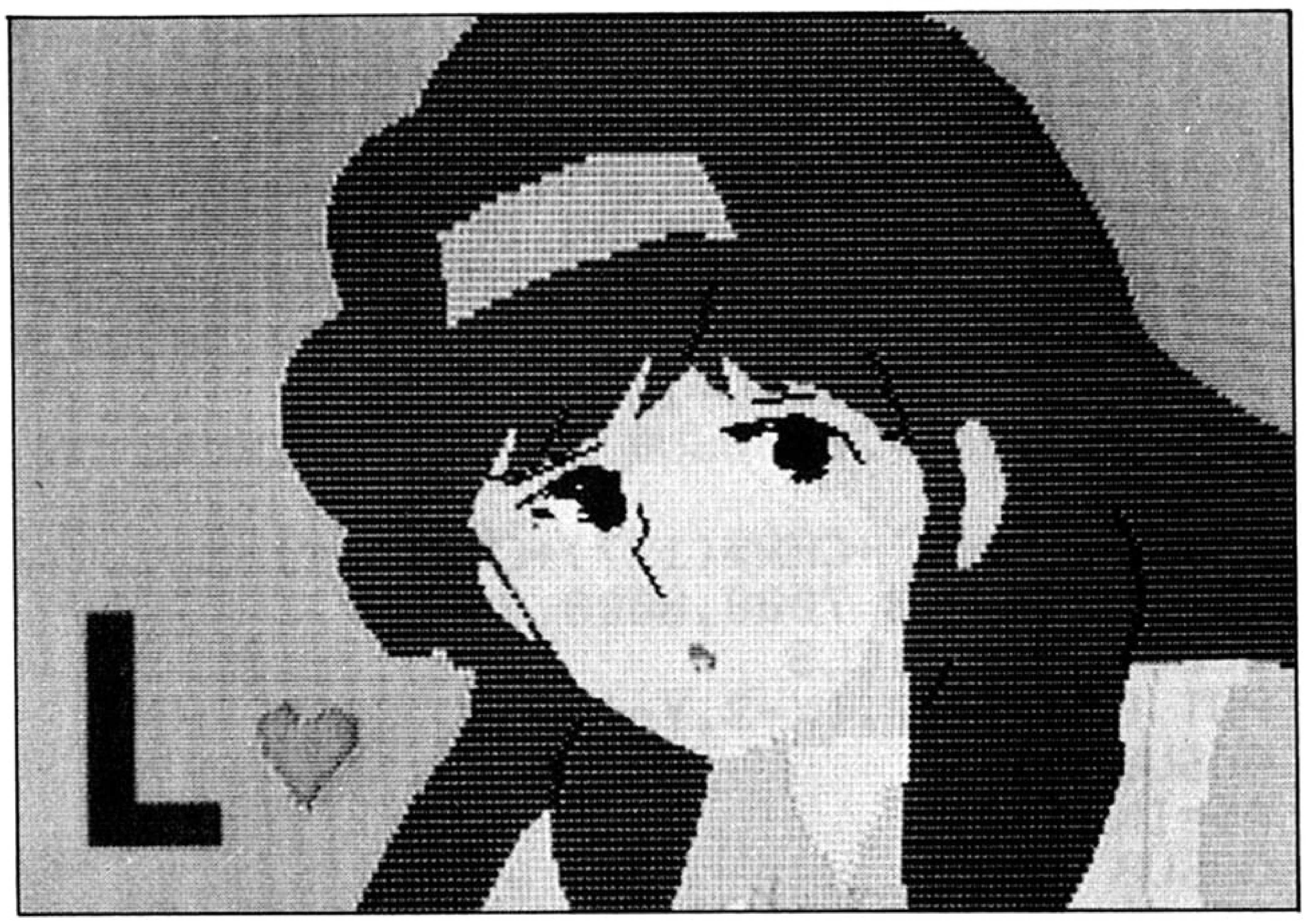
```
100 CLEAR 300:DEFINT A-Z
110 MX=2000:DIM X(MX),Y(MX),DX(8),DY(8)
120 INPUT "USE STICK or KEY (j/k) ";A$
130 IF A$="j" OR A$="J" THEN ST=1:GOTO 160
140 IF A$="k" OR A$="K" THEN ST=0:GOTO 160
150 GOTO 120
160 SCREEN 2,0:COLOR 15,4,15:CLS
170 A$=STRING$(3,&H10)+CHR$(&HFE)+STRING$(3,&H10)+CHR$(0)
180 SPRITE$(0)=A$
190 SPRITE$(1)=A$
200 FOR I=1 TO 8
210 READ DX(I),DY(I)
220 NEXT I
230 DATA 0,-1,1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1
240 MN$=""
250 FOR I=1 TO 34
260 READ A$
270 MN$=MN$+CHR$(VAL("&H"+A$))
280 NEXT
290 DATA 21,00,00,11,00,00,01,00,18,cd,4a,00,f5,e6,0f,ba,20
300 DATA 08,f1,e6,f0,b3,cd,4d,00,3e,f1,0b,23,78,b1,20,e8,c9
310 X=128:Y=96:I=0:C=15:B=4
320 ON KEY GOSUB 8000,9000,7000
330 GOSUB 10000
340 PUT SPRITE 0,(X-3,Y-4),15
350 A$=INKEY$
360 IF A$="b" THEN GOSUB 1000
370 IF A$="c" THEN GOSUB 2000
380 IF A$="l" THEN GOSUB 3000
390 IF A$=CHR$(11) THEN GOSUB 4000
400 IF A$=CHR$(127) THEN GOSUB 5000
410 IF STRIG(ST)=-1 THEN GOSUB 6000
420 PUT SPRITE 0,(X-3,Y-4),1
430 X=X+DX(STICK(ST))
440 Y=Y+DY(STICK(ST))
450 IF X<0 THEN X=255
460 IF X>255 THEN X=0
470 IF Y<0 THEN Y=191
480 IF Y>191 THEN Y=0
490 GOTO 340
1000 REM -- ﾊﾞｯｸ ｶﾗｰ --
1010 B1=B+1:IF B1>15 THEN B1=0
1020 J=BASE(11)
1030 MID$(MN$,2,2)=CHR$(J MOD 256)+CHR$(J ¥ 256)
```

```
1040 MID$(MN$,5,2)=CHR$(B1)+CHR$(B)
1050 J=VARPTR(MN$)+1
1060 DEF USR=PEEK(J)+PEEK(J+1)*256
1070 J=USR(0)
1080 B=B1
1090 COLOR C,B,C
1100 RETURN
2000 REM -- ﾌｫｱ ｶﾗｰ  --
2010 C=C+1:IF C>15 THEN C=0
2020 COLOR C,B,C
2030 RETURN
3000 REM --  line  --
3010 IF I>=MX THEN GOSUB 12000:GOTO 3080
3020 X(I)=-1
3030 I=I+1
3040 X(I)=X+C*256:Y(I)=Y
3050 X1=X:Y1=Y
3060 I=I+1
3070 PUT SPRITE 1,(X-3,Y-4),C
3080 RETURN
4000 REM --  かきなおし --
4010 GOSUB 11000
4020 CLS
4030 PUT SPRITE 1,(0,209)
4040 IF I=0 THEN 4120
4050 FOR J=0 TO I-1
4060 A=X(J):AY=Y(J)
4070 IF A=-1 THEN LF=1:GOTO 4110
4080 AX=A MOD 256:AC=A ¥ 256
4090 IF LF=1 THEN DRAW"BM=AX;,=AY;" ELSE LINE-(AX,AY),AC
4100 LF=0
4110 NEXT J
4120 PUT SPRITE 1,(X1-3,Y1-4),C
4130 GOTO 10000
5000 REM --  ﾊﾞｯｸ  --
5010 IF I=0 THEN 5110
5020 IF I=1 THEN 5180
5030 I=I-1
5040 IF X(I-1)=-1 THEN 5120
5050 X1=X(I-1) MOD 256
5060 Y1=Y(I-1)
5070 C=X(I-1) ¥ 256
5080 LINE(X1,Y1)-(X(I) MOD 256,Y(I)),B
5090 PUT SPRITE 1,(X1-3,Y1-4),C
5100 COLOR C,B,C
5110 RETURN
```

```
5120 IF I=1 THEN 5180
5130 I=I-1
5140 X1=X(I-1) MOD 256
5150 Y1=Y(I-1)
5160 C=X(I-1) ¥ 256
5170 GOTO 5090
5180 CLS
5190 PUT SPRITE 1,(0,209)
5200 I=0:X1=0:Y1=0
5210 RETURN
6000 REM -- けいそ゛く てん --
6005 IF I=0 THEN 3000
6010 IF X=X1 AND Y=Y1 THEN 6080
6020 IF I>MX THEN GOSUB 12000:GOTO 6080
6030 X(I)=X+C*256:Y(I)=Y
6040 I=I+1
6050 LINE(X1,Y1)-(X,Y),C
6060 PUT SPRITE 1,(X-3,Y-4)
6070 X1=X:Y1=Y
6080 RETURN
7000 REM -- BASIC フ゜ロク゛ラム --
7010 GOSUB 11000
7020 COLOR 15,4,7:SCREEN 1
7030 PRINT:PRINT "        BASIC セーフ゛"
7040 LOCATE 5,12
7050 A$=""
7060 INPUT "BASIC FILE NAME";A$
7070 IF A$="" THEN 7370
7080 OPEN A$ FOR OUTPUT AS #1
7090 PRINT#1,"10 defint a-z
7100 PRINT#1,"20 color 15,"+STR$(B)+",4:screen 2"
7110 PRINT#1,"30 read x,y"
7120 PRINT#1,"40 if x=-1 then 70
7130 PRINT#1,"50 line-(x mod 256,y),x ¥ 256"
7140 PRINT#1,"60 goto 30"
7150 PRINT#1,"70 if y=-1 then 70"
7160 PRINT#1,"80 read x,y
7170 PRINT#1,"90 if x=-1 then 70
7180 PRINT#1,"100 x=x mod 256
7190 PRINT#1,"110 draw"+CHR$(34)+"bm=x;,=y;"+CHR$(34)
7200 PRINT#1,"120 goto 30
7210 A=0:L=1000
7220 FOR J=0 TO I-1
7230 IF A=0 THEN A$=STR$(L)+" data "
7240 A$=A$+STR$(X(J))+","+STR$(Y(J))
7250 IF A=3 THEN 7290
```

```
7260 A$=A$+","
7270 A=A+1
7280 GOTO 7320
7290 PRINT#1,A$
7300 L=L+10
7310 A=0
7320 NEXT J
7330 IF A=0 THEN A$=STR$(L)+" data "
7340 A$=A$+"-1,-1"
7350 PRINT#1,A$
7360 CLOSE
7370 COLOR C,B,C:SCREEN 2
7380 GOSUB 4000
7390 GOTO 10000
8000 REM --  ﾛｰﾄﾞ  --
8010 GOSUB 11000
8020 COLOR 15,4,7:SCREEN 1
8030 PRINT:PRINT "           ﾃﾞｰﾀ ﾛｰﾄﾞ"
8040 LOCATE 5,12
8050 A$=""
8060 INPUT "File Name";A$
8070 IF A$="" THEN 8170
8080 A$=LEFT$(A$+"      ",6)
8090 OPEN A$+"i" FOR INPUT AS #1
8100 INPUT #1,I,B,AX,AY
8110 CLOSE #1
8120 BLOAD A$+"x",VARPTR(X(0))-AX
8130 BLOAD A$+"y",VARPTR(Y(0))-AY
8140 X1=X(I-1) MOD 256
8150 Y1=Y(I-1)
8160 C=X(I-1) ¥ 256
8170 COLOR C,B,C:SCREEN 2
8180 GOSUB 4000
8190 GOTO 10000
9000 REM --  ｾｰﾌﾞ  --
9010 GOSUB 11000
9020 COLOR 15,4,7:SCREEN 1
9030 PRINT:PRINT "           ﾃﾞｰﾀ ｾｰﾌﾞ"
9040 LOCATE 5,12
9050 A$="":AX=0:AY=0
9060 INPUT "File Name";A$
9070 IF A$="" THEN 9150
9080 A$=LEFT$(A$+"      ",6)
9090 OPEN A$+"i" FOR OUTPUT AS #1
9100 AX=VARPTR(X(0)):AY=VARPTR(Y(0))
9110 PRINT #1,I,B,AX,AY
```

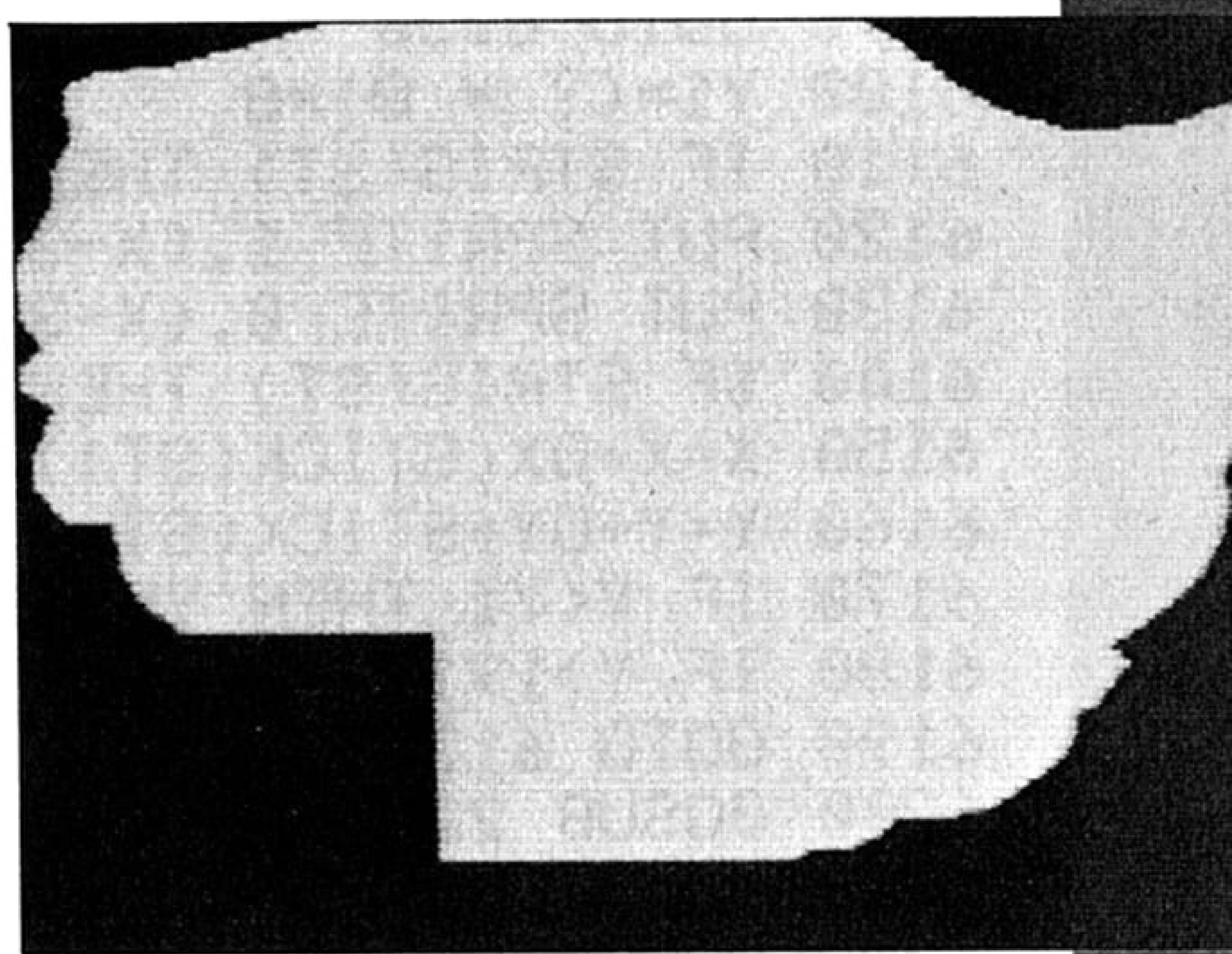
```
9120 CLOSE #1
9130 BSAVE A$+"x",AX,AX+I*2-1
9140 BSAVE A$+"y",AY,AY+I*2-1
9150 COLOR C,B,C:SCREEN 2
9160 GOSUB 4000
9170 GOTO 10000
10000 REM ==  ｷｰ･ｵﾝ  ==
10010 KEY(1) ON
10020 KEY(2) ON
10030 KEY(3) ON
10040 RETURN
11000 REM ==  ｷｰ･ｵﾌ  ==
11010 KEY(1) OFF
11020 KEY(2) OFF
11030 KEY(3) OFF
11040 RETURN
12000 '-- Out of Memory --
12010 SCREEN 1
12020 PRINT "Out of Memory"
12030 PRINT
12040 PRINT "Hit RETURN Key";
12050 DM$=INPUT$(1)
12060 IF DM$<>CHR$(13) THEN 12050
12070 SCREEN 2
12080 GOSUB 4000
12090 RETURN
```



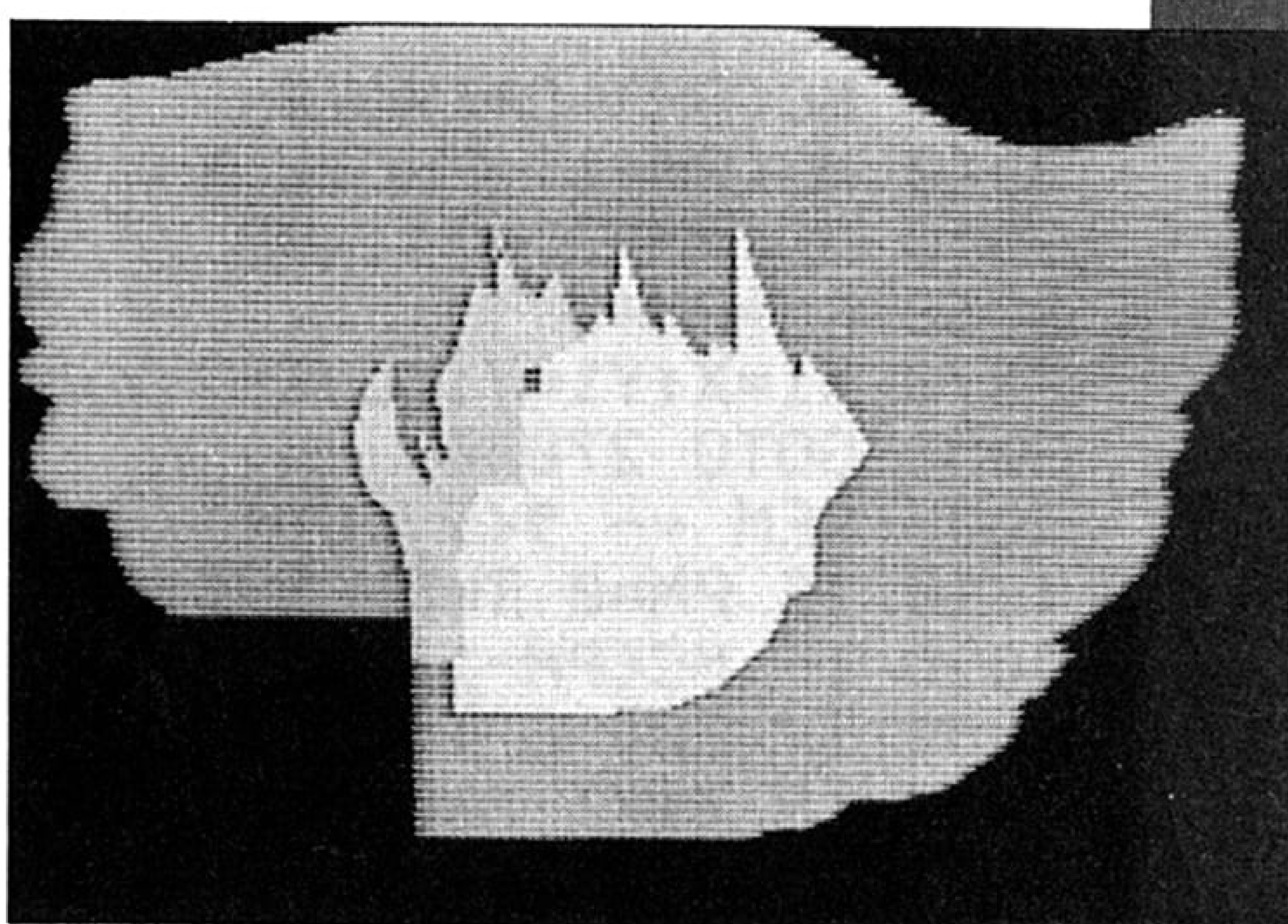
アニメ・キャラクタ用ツール　GT-2

```
100 CLEAR 500,&HF1FF:DEFINT A-Z
110 DEF FND(X,Y)=VAL("&h"+HEX$(X)+"00")+Y
120 DEF FNX(D)=PEEK(VARPTR(D)+1)
130 DEF FNY(D)=PEEK(VARPTR(D))
140 MX=1500:DIM MC(MX),DX(8),DY(8)
150 INPUT "JOY STICK or KEY (j/k) ";A$
160 IF A$="J" OR A$="j" THEN ST=1:GOTO 190
170 IF A$="K" OR A$="k" THEN ST=0:GOTO 190
180 GOTO 150
190 FOR I=1 TO 8:READ DX(I),DY(I):NEXT
200 DATA 0,-1,1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1
210 X=128:Y=96:C=15:B=12:SCREEN 2,0,0
220 FOR I=1 TO 3
230 KEY I,""
240 NEXT
250 A$=STRING$(2,16)
260 A$=A$+CHR$(0)+CHR$(&HC6)+CHR$(0)+A$+CHR$(0)
270 SPRITE$(0)=A$:SPRITE$(1)=A$
280 MN$=""
290 FOR I=&HF200 TO &HF29F
300 READ A$:POKE I,VAL("&h"+A$)
310 NEXT
320 DATA 21,00,00,11,00,00,01,00,00,CD,4A,00,CD,22,F2,CD
330 DATA 22,F2,CD,4D,00,23,7B,07,07,07,07,5F,0B,78,B1,20
340 DATA E8,C9,F5,E6,0F,BA,20,0D,F1,D5,F5,7B,E6,0F,5F,F1
350 DATA E6,F0,B3,D1,F5,F1,07,07,07,07,C9,00,00,00,00,00
360 DATA 21,00,20,11,00,00,01,00,18,CD,4A,00,F5,E6,0F,BA
370 DATA 20,08,F1,E6,F0,B3,CD,4D,00,3E,F1,0B,23,78,B1,20
380 DATA E8,C9,11,00,3E,21,00,00,06,00,CD,86,F2,21,00,20
390 DATA CD,86,F2,C9,21,00,3E,11,00,00,06,00,CD,86,F2,11
400 DATA 00,20,CD,86,F2,C9,CD,4A,00,EB,CD,4D,00,EB,23,13
410 DATA 10,F4,C9,00,00,00,00,00,00,00,00,00,00,00,00,00
420 PM=0:X1=0:Y1=0:CU=15
430 ON INTERVAL=20 GOSUB 26000:INTERVAL ON
440 CD=0
450 CM$="bclpt?"+CHR$(127)+CHR$(11)
460 ON KEY GOSUB 7000,8000,9000
470 COLOR C,B,C:CLS
480 PUT SPRITE 0,(X-3,Y-4),CU
490 GOSUB 30000
500 A$=INKEY$
510 GOSUB 31000
520 IF A$="" THEN 540
```

```
530 ON INSTR(CM$,A$) GOSUB 1000,2000,3000,5000
,6000,10000,11000,12000
540 IF STRIG(ST) THEN GOSUB 4000
550 X=X+DX(STICK(ST))
560 Y=Y+DY(STICK(ST))
570 X=X AND 255
580 IF Y<0 THEN Y=191
590 IF Y>191 THEN Y=0
600 GOTO 480
1000 REM -- ﾊﾞｯｸ ｶﾗｰ --
1010 GOSUB 22000
1020 A$="BACK COLOR = ?"
1030 GOSUB 25000
1040 IF A$="" THEN GOSUB 23000:RETURN
1050 B1=VAL(A$) AND 15
1060 GOSUB 23000
1070 POKE &HF244,B1:POKE &HF245,B
1080 DEF USR=&HF240
1090 I=USR(0):B=B1
1100 COLOR C,B,C
1110 RETURN
2000 REM -- ﾌｫｱ ｶﾗｰ --
2010 C=C+1 AND 15
2020 COLOR C,B,C
2030 RETURN
3000 REM -- ﾗｲﾝ (してん) --
3010 IF PM>MX-3 THEN 20000
3020 CD=1
3030 MC(PM)=C+&H1E0
3040 MC(PM+1)=FND(X,Y)
3050 MC(PM+2)=511
3060 PM=PM+3
3070 X1=X:Y1=Y:CO=C
3080 GOTO 27000
4000 REM -- ﾗｲﾝ (けいぞくてん) --
4010 IF PM=0 THEN 3000
4020 IF MC(PM-1)<>511 THEN 3000
4030 IF CO<>C THEN 3000
4040 IF X=X1 AND Y=Y1 THEN RETURN
4050 IF PM>MX-1 THEN 20000
4060 MC(PM-1)=FND(X,Y)
4070 MC(PM)=511
4080 PM=PM+1
4090 LINE(X1,Y1)-(X,Y),C
4100 X1=X:Y1=Y
4110 GOTO 27000
```



```
5000 REM --  ﾍﾟｲﾝﾄ  --
5010 IF PM>MX-3 THEN 20000
5020 IF MC(PM-1)=255 AND CO=C THEN PM=PM-2:GOTO 5040
5030 MC(PM)=C+&HE0
5040 MC(PM+1)=FND(X,Y)
5050 MC(PM+2)=255
5060 PM=PM+3
5070 CD=0:GOSUB 28000
5080 PAINT(X,Y),C
5090 CO=C:RETURN
6000 REM -- ｶﾗｰﾁｪﾝｼﾞ --
6010 GOSUB 28000
6020 ON INTERVAL=10 GOSUB 26000:INTERVAL ON
6030 PUT SPRITE 0,(X-3,(Y ¥ 8)*8-4),CU
6040 IF STRIG(ST) THEN 6100
6050 X=X+DX(STICK(ST)) AND 255
6060 Y=Y+DY(STICK(ST))
6070 IF Y<0 THEN Y=191
6080 IF Y>191 THEN Y=0
6090 GOTO 6030
6100 Y1=(Y ¥ 8)*8
6110 IF STRIG(ST) THEN 6110
6120 PUT SPRITE 1,(X-3,Y1-4),C
6130 PUT SPRITE 0,(X-3,(Y ¥ 8)*8+3),CU
6140 IF STRIG(ST) THEN 6200
6150 X=X+DX(STICK(ST)) AND 255
6160 Y=Y+DY(STICK(ST))
6170 IF Y<Y1 THEN Y=Y1
6180 IF Y>191 THEN Y=191
6190 GOTO 6120
6200 GOSUB 22000
6210 A$="COLOR 1 ="
6220 GOSUB 25000
6230 IF A$="" THEN 6520
6240 C1=VAL(A$)
6250 A$="COLOR 2 ="
6260 GOSUB 24000
6270 LINE(200,0)-(255,7),C1,BF
6280 DRAW"bm92,0"
6290 GOSUB 25020
6300 IF A$="" THEN 6520
6310 C2=VAL(A$)
6320 A$="Ok ? (y/n) "
6330 GOSUB 24000
6340 LINE(200,0)-(255,7),C1,BF
6350 FOR I=1 TO 7 STEP 2
```



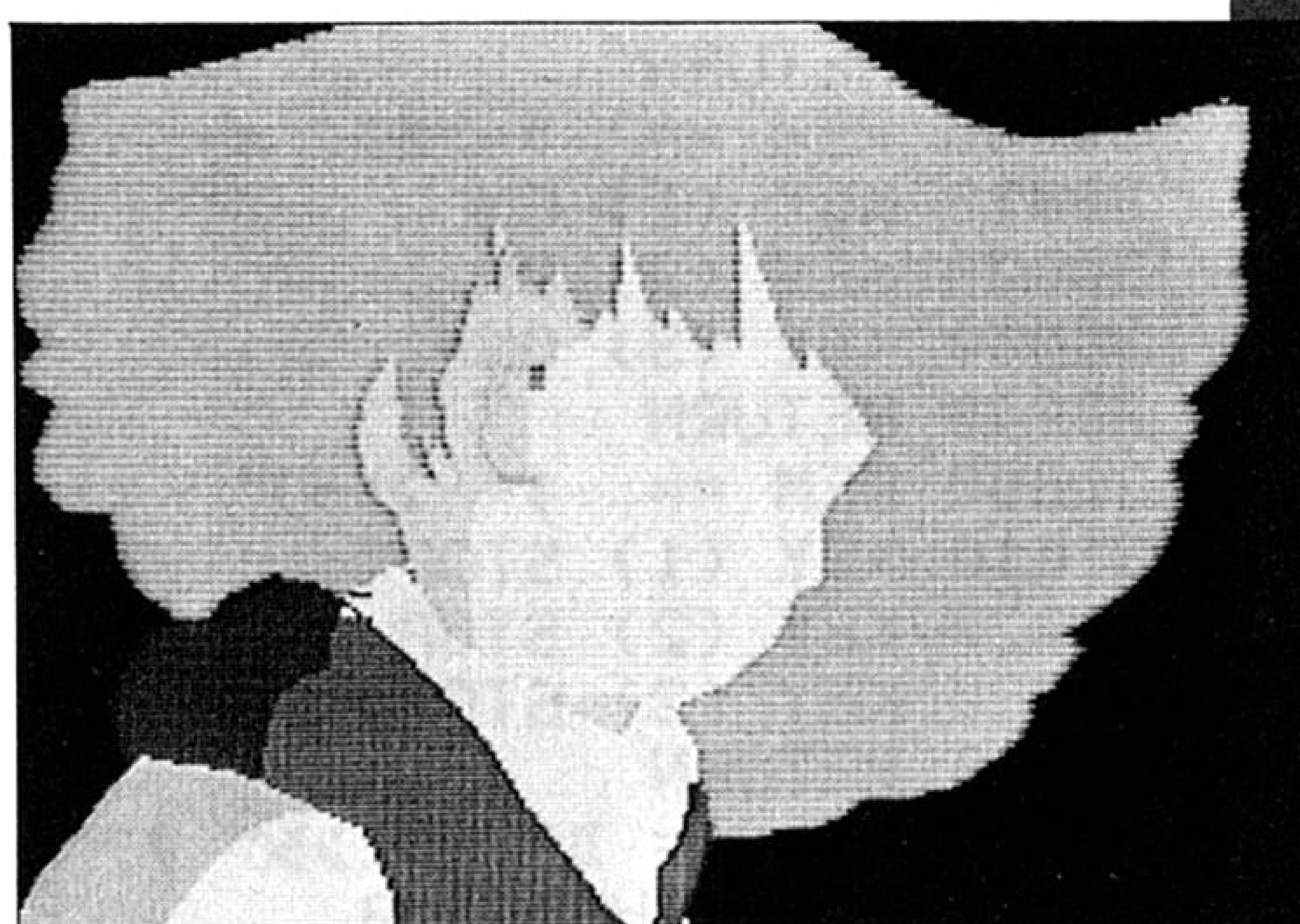
```
6360 LINE(200,I)-(255,I),C2,BF
6370 NEXT
6380 DRAW"bm108,0"
6390 GOSUB 25020
6400 IF A$="N" OR A$="n" THEN 6210
6410 IF A$<>"Y" AND A$<>"y" THEN 6320
6420 GOSUB 23000
6430 IF PM>MX-2 THEN 20000
6440 S=Y1 ¥ 8:E=(Y-Y1) ¥ 8+1
6450 MC(PM)=FND(C1*16+C2,C)
6460 MC(PM+1)=FND(S,E)
6470 PM=PM+2
6480 GOSUB 32000
6490 CD=0:GOSUB 28000
6500 ON INTERVAL=30 GOSUB 26000:INTERVAL ON
6510 RETURN
6520 GOSUB 23000:GOTO 6490
7000 REM --  ﾛｰﾄﾞ  --
7010 GOSUB 29000
7020 GOSUB 22000
7030 A$="LOAD FILE NAME "
7040 GOSUB 25000
7050 IF A$="" THEN 23000
7060 GOSUB 21000
7070 OPEN A$+"i"FOR INPUT AS #1
7080 INPUT#1,B,PM,Y1
7090 CLOSE
7100 BLOAD A$+"m",VARPTR(MC(0))-Y1
7110 GOSUB 23000
7120 COLOR ,B
7130 GOTO 12000
8000 REM --  ｾｰﾌﾞ  --
8010 GOSUB 29000
8020 GOSUB 22000
8030 A$="SAVE FILE NAME "
8040 GOSUB 25000
8050 IF A$="" THEN 8120
8060 GOSUB 21000
8070 OPEN A$+"i" FOR OUTPUT AS #1
8080 Y1=VARPTR(MC(0))
8090 PRINT#1,B,PM,Y1
8100 CLOSE
8110 BSAVE A$+"m",Y1,Y1+PM*2
8120 GOSUB 23000
8130 RETURN
```

```
9000 REM -- BASIC ﾌﾟﾛｸﾞﾗﾑ --
9010 GOSUB 29000
9020 GOSUB 22000
9030 A$="BASIC PROG. NAME "
9040 GOSUB 25000
9050 IF A$="" THEN 23000
9060 GOSUB 21000
9070 OPEN A$ FOR OUTPUT AS #1
9080 PRINT#1,"10 clear 300,&hf1ff:gosub 300"
9090 PRINT#1,"20 color 15,"+STR$(B)+",4:screen 2"
9100 PRINT#1,"30 a=0:d=varptr(a)"
9110 PRINT#1,"40 read a"
9120 PRINT#1,"50 if a=-1 then 50"
9130 PRINT#1,"60 b=a and &h1f0"
9140 PRINT#1,"70 if b=&he0 then 100"
9150 PRINT#1,"80 if b=&h1e0 then 150"
9160 PRINT#1,"90 goto 230"
9170 PRINT#1,"100 c=a-b"
9180 PRINT#1,"110 read a"
9190 PRINT#1,"120 if a=255 then 40"
9200 PRINT#1,"130 paint(peek(d+1),peek(d)),c"
9210 PRINT#1,"140 goto 110"
9220 PRINT#1,"150 c=a-b"
9230 PRINT#1,"160 read a"
9240 PRINT#1,"170 b=peek(d):a=peek(d+1)"
9250 PRINT#1,"180 draw"+CHR$(34)+"bm=a;,=b;"+CHR$(34)
9260 PRINT#1,"190 read a"
9270 PRINT#1,"200 if a=511 then 40"
9280 PRINT#1,"210 line-(peek(d+1),peek(d)),c"
9290 PRINT#1,"220 goto 190"
9300 PRINT#1,"230 poke &hf204,peek(d+1)"
9310 PRINT#1,"240 poke &hf205,peek(d)"
9320 PRINT#1,"250 read a"
9330 PRINT#1,"260 poke &hf202,peek(d+1)+&h20"
9340 PRINT#1,"270 poke &hf208,peek(d)"
9350 PRINT#1,"280 a=usr(0)"
9360 PRINT#1,"290 goto 40"
9370 PRINT#1,"300 defint a-z"
9380 PRINT#1,"310 for a=&hf200 to &hf23a"
9390 PRINT#1,"320 read a$:poke a,val(";CHR$(34);"&H"
;CHR$(34);"+a$)"
9400 PRINT#1,"330 next"
9410 PRINT#1,"340 def usr=&hf200"
9420 PRINT#1,"350 return"
9430 PRINT#1,"360 rem --  ﾏｼﾝｺﾞ ﾃﾞｰﾀ  --"
9440 PRINT#1,"370 DATA 21,00,00,11,00,00,01,00,00,CD
,4A,00,CD,22,F2,CD"
```

```
9450 PRINT#1,"380 DATA 22,F2,CD,4D,00,23,7B,07,07,07,07,
5F,0B,78,B1,20"
9460 PRINT#1,"390 DATA E8,C9,F5,E6,0F,BA,20,0D,F1,D5,F5,
7B,E6,0F,5F,F1"
9470 PRINT#1,"400 DATA E6,F0,B3,D1,F5,F1,07,07,07,07,C9"
9480 PRINT#1,"1000 rem ---   ";A$;"DATA  ---"
9490 L=1010:P=0
9500 FOR I=0 TO PM-1
9510 IF P=0 THEN A$=MID$(STR$(L),2)+"data "
9520 A1$=STR$(MC(I))
9530 IF ASC(A1$)=32 THEN A1$=MID$(A1$,2)
9540 A$=A$+A1$
9550 IF P=5 THEN 9590
9560 A$=A$+","
9570 P=P+1
9580 GOTO 9610
9590 PRINT#1,A$
9600 L=L+10:P=0
9610 NEXT
9620 IF P=0 THEN A$=MID$(STR$(L),2)+"data "
9630 PRINT#1,A$+"-1"
9640 CLOSE
9650 GOTO 23000
10000 REM --  じょうほう  --
10010 GOSUB 28000
10020 GOSUB 22000
10030 A$="POINT ("+STR$(X)+","+STR$(Y)+") COLOR"+STR$(PO
INT(X,Y))
10040 GOSUB 24000
10050 IF INPUT$(1)=CHR$(13) THEN 10150
10060 A$="LAST POINT ("+STR$(X1)+","+STR$(Y1)+")"
10070 GOSUB 24000
10080 IF INPUT$(1)=CHR$(13) THEN 10150
10090 A$="PEN COLOR"+STR$(C)+" : BACK"+STR$(B)
10100 GOSUB 24000
10110 IF INPUT$(1)=CHR$(13) THEN 10150
10120 A$="USE"+STR$(PM)+" : FREE"+STR$(MX-PM)
10130 GOSUB 24000
10140 IF INPUT$(1)=CHR$(13) THEN 10150
10150 GOSUB 27000
10160 GOTO 23000
11000 REM --  ﾊﾞｯｸ  --
11010 IF PM=0 THEN RETURN
11020 IF PM=1 THEN PM=0:CLS:RETURN
11030 PM=PM-1
11040 IF MC(PM)<>255 THEN 11070
```

```
11050 IF (MC(PM-2) AND &H1F0)=&HE0 THEN PM=PM-2:GOTO 11090
11060 MC(PM-1)=255:GOTO 11090
11070 IF MC(PM)=511 THEN 11110
11080 PM=PM-1
11090 GOSUB 12000
11100 RETURN
11110 IF (MC(PM-2) AND &H1F0)=&H1E0 THEN 11160
11120 X1=FNX(MC(PM-2)):Y1=FNY(MC(PM-2))
11130 LINE(X1,Y1)-(FNX(MC(PM-1)),FNY(MC(PM-1))),B
11140 MC(PM-1)=511
11150 GOTO 11210
11160 PM=PM-2
11170 IF PM=0 THEN 11190
11180 IF MC(PM-1)=511 THEN 11200
11190 CD=0:GOTO 28000
11200 X1=FNX(MC(PM-2)):Y1=FNY(MC(PM-2))
11210 CD=1:GOTO 27000
12000 REM --  かきなおし  --
12010 IF PM=0 THEN RETURN
12020 INTERVAL OFF
12030 CLS
12040 P=0:L=0
12050 FOR I=0 TO PM-1
12060 A=MC(I)
12070 IF P=1 THEN 12230
12080 IF P=2 THEN 12180
12090 IF L=1 THEN 12260
12100 IF L=2 THEN 12290
12110 IF (A AND &H1F0)=&HE0 THEN C=A-&HE0:P=1:GOTO 12320
12120 IF (A AND &H1F0)=&H1E0 THEN C=A-&H1E0:L=1:GOTO 12320
12130 C1=FNX(A) ¥ 16
12140 C2=FNX(A) MOD 16
12150 C=FNY(A)
12160 P=2
12170 GOTO 12320
12180 S=FNX(A)
12190 E=FNY(A)
12200 GOSUB 32000
12210 P=0
12220 GOTO 12320
12230 IF A=255 THEN P=0:GOTO 12320
12240 PAINT(FNX(A),FNY(A)),C
12250 GOTO 12320
12260 X1=FNX(A):Y1=FNY(A)
12270 DRAW"bm=x1;,=y1;"
12280 L=2:GOTO 12320
```

```
12290 IF A=511 THEN L=0:GOTO 12320
12300 X1=FNX(A):Y1=FNY(A)
12310 LINE-(X1,Y1),C
12320 NEXT
12330 INTERVAL ON
12340 COLOR C,B,C
12350 RETURN
20000 REM ==  エラー  ==
20010 GOSUB 22000
20020 A$="Out of Memory "
20030 GOSUB 24000
20040 BEEP:BEEP:BEEP
20050 OPEN"GRP:"AS #1
20060 PRINT#1," Hit RETURN _"
20070 CLOSE
20080 IF INKEY$<>CHR$(13) THEN 20080
20090 GOTO 23000
21000 REM == ファイルネーム ==
21010 A$=LEFT$(A$+"        ",6+INSTR(A$,":"))
21020 RETURN
22000 REM ==  たいひ  ==
22010 DEF USR=&HF262
22020 I=USR(0)
22030 RETURN
23000 REM ==  ふっき  ==
23010 DEF USR=&HF274
23020 I=USR(0)
23030 RETURN
24000 REM == メッセージ ==
24010 LINE(0,0)-(255,7),4,BF
24020 OPEN"GRP:"AS #1
24030 DRAW"bm20,0"
24040 COLOR 15
24050 PRINT#1,A$;
24060 CLOSE
24070 COLOR C
24080 RETURN
25000 REM == インプット ==
25010 GOSUB 24000
25020 OPEN"GRP:"AS #1
25030 COLOR 15
25040 IF INKEY$<>"" THEN 25040
25050 A$=""
25060 PRINT#1,CHR$(95);
25070 I$=INPUT$(1)
25080 IF I$=CHR$(13) THEN CLOSE:COLOR C:RETURN
```



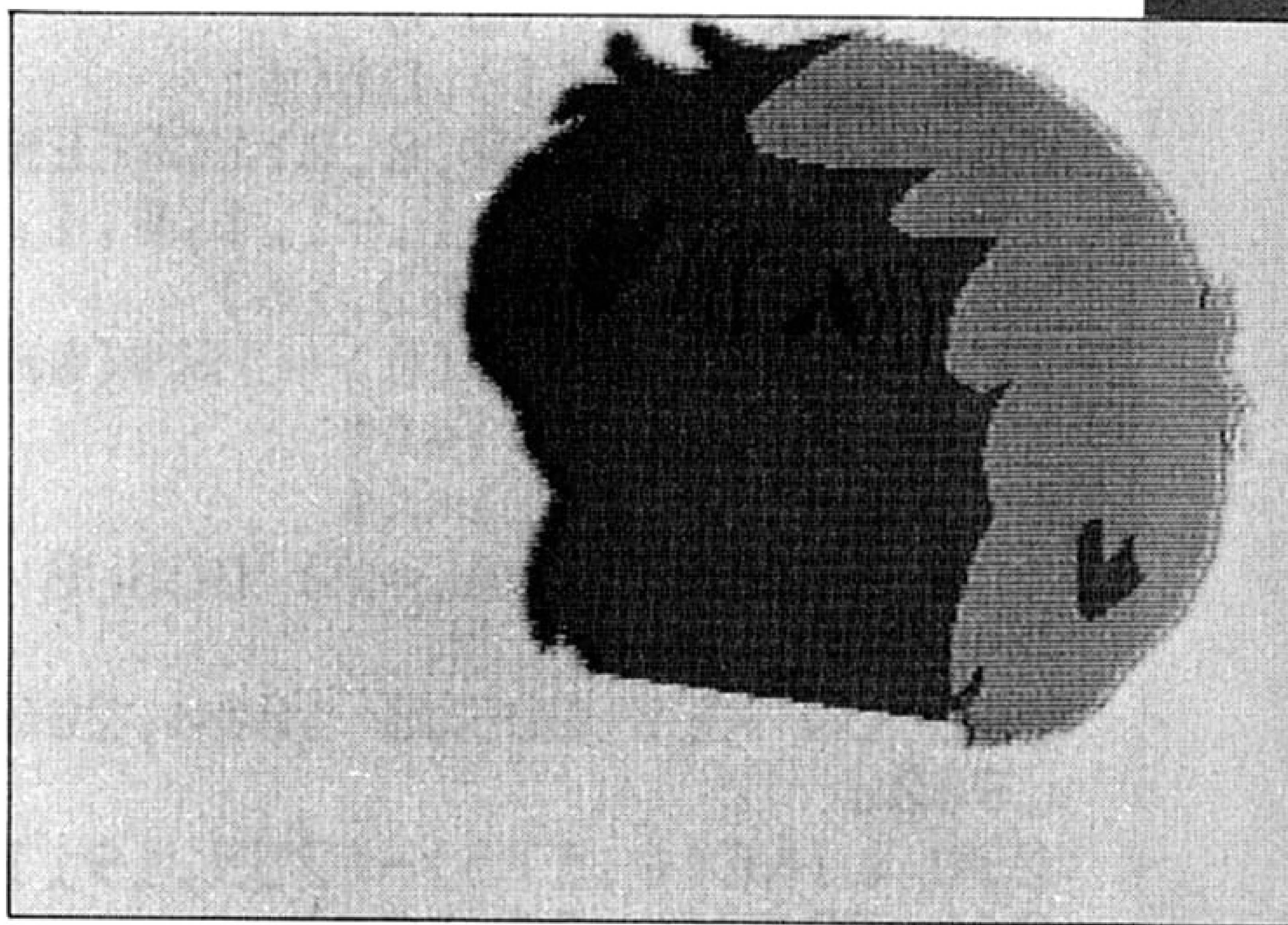
```
25090 IF I$=CHR$(8) THEN LINE STEP(-1,7)-STEP
(-LEN(A$)*8-7,-7),4,BF:GOTO 25050
25100 IF LEN(A$)>10 THEN 25070
25110 IF I$<" " THEN 25070
25120 A$=A$+I$
25130 LINE STEP(-1,7)-STEP(-7,-7),4,BF
25140 PRINT#1,I$;:GOTO 25060
26000 REM == カーソル ==
26010 CU=16-CU
26020 INTERVAL ON
26030 RETURN
27000 REM == サブカーソル オン ==
27010 IF CD THEN PUT SPRITE 1,(X1-3,Y1-4),C
27020 RETURN
28000 REM == サブカーソル オフ ==
28010 PUT SPRITE 1,(0,209)
28020 RETURN
29000 REM ==  キー・オフ  ==
29010 KEY (1) OFF
29020 KEY (2) OFF
29030 KEY (3) OFF
29040 RETURN
30000 REM ==  キー・オン  ==
30010 KEY (1) ON
30020 KEY (2) ON
30030 KEY (3) ON
30040 RETURN
31000 REM ==  キー・ストップ  ==
31010 KEY (1) STOP
31020 KEY (2) STOP
31030 KEY (3) STOP
31040 RETURN
32000 REM == カラー・チェンジ サブ ==
32010 POKE &HF201,0
32020 POKE &HF202,S+&H20
32030 POKE &HF204,C1*16+C2
32040 POKE &HF205,C
32050 POKE &HF207,0
32060 POKE &HF208,E
32070 DEF USR=&HF200
32080 U=USR(0)
32090 RETURN
```



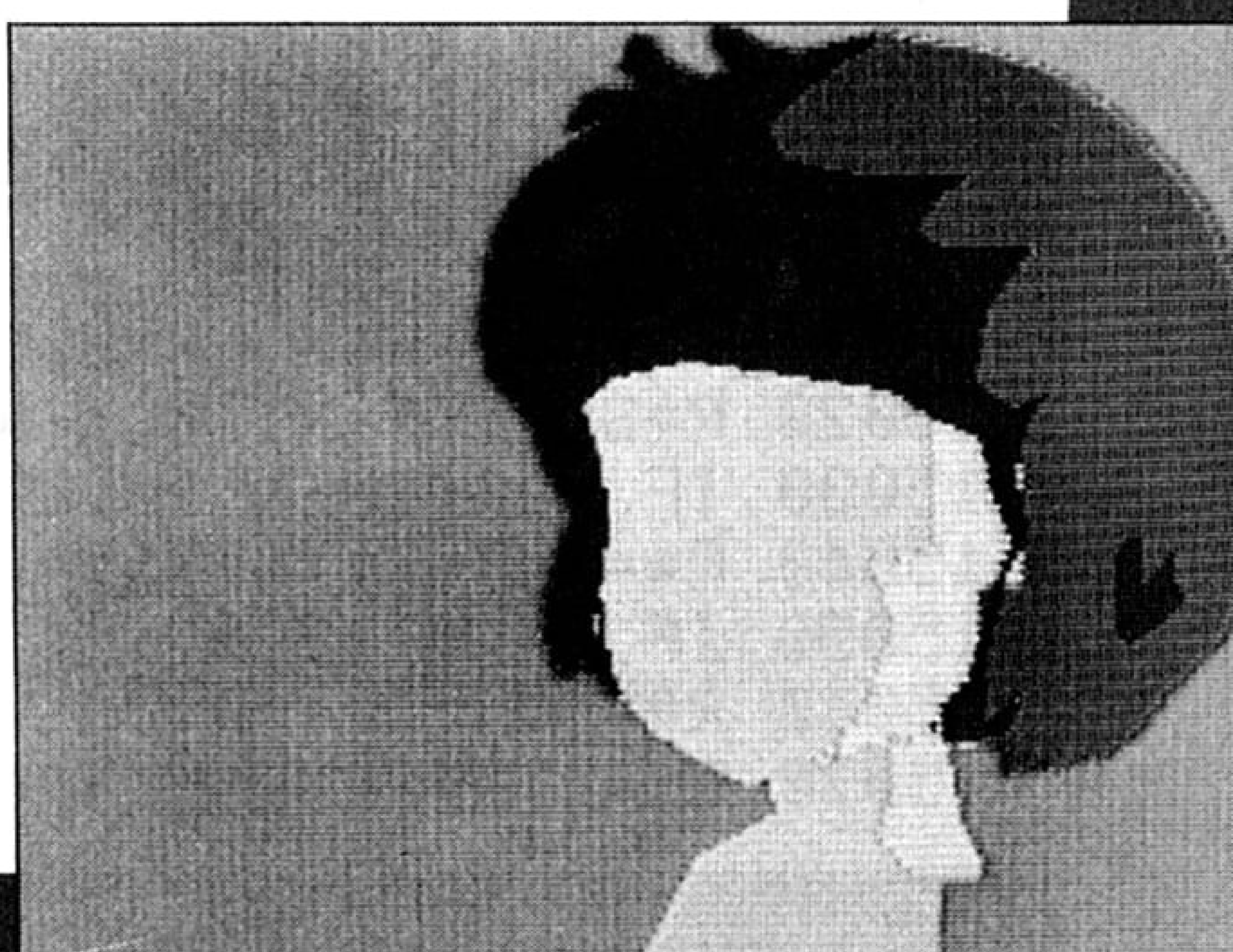
「お絵描きツール」本格版　GT-3

```
100 DEFINT A-Z
110 DIM DX(8),DY(8)
120 FOR I=1 TO 8
130 READ DX(I),DY(I)
140 NEXT:COLOR,0,0:SCREEN 2,0
150 DATA 0,-1,1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1
160 A$=STRING$(2,16)
170 A$=A$+CHR$(0)+CHR$(&HC6)+CHR$(0)+A$
180 SPRITE$(0)=A$
190 SPRITE$(1)=A$
200 ON INTERVAL=20 GOSUB 29000
210 INTERVAL ON
220 ON KEY GOSUB 1000,2000,3000,4000,5000,6000,7000,8000
,9000
230 LINE(0,175)-(255,191),4,BF
240 DRAW"bm24,184"
250 A$="ﾓｰﾄﾞ:          ふとさ:"
260 GOSUB 26000
270 X=128:Y=96:MX=255:MY=174:NX=0:NY=0
280 ST=0:CU=15:CS=-1:M=1:F=1:C=15
290 GOSUB 28000
300 GOSUB 21000
310 GOSUB 22000
320 GOSUB 20000
330 ON M GOSUB 10000,11000,12000,13000,14000,15000
340 GOTO 300
1000 REM -- ﾓｰﾄﾞ --
1010 GOSUB 20000
1020 DRAW"bm32,176"
1030 A$="DOT LIN BOX FIL PAI CIR"
1040 GOSUB 26000
1050 P0=M:P1=32:P2=-2:P3=175:P4=25:P5=8:P6=1:P7=6
1060 GOSUB 25000
1070 M=P0
1080 GOSUB 27000
1090 FF=-1
1100 GOSUB 24000
1110 GOSUB 21000
1120 GOTO 28000
2000 REM --  ﾛｰﾄﾞ  --
2010 GOSUB 20000
2020 DRAW"bm24,176"
2030 A$="ﾛｰﾄﾞ ﾌｧｲﾙめい "
```

```
2040 GOSUB 30000
2050 IF I$=CHR$(27) THEN 2200
2060 OPEN"CAS:"+A$ FOR INPUT AS #1
2070 FOR I=0 TO 5631
2080 A=ASC(INPUT$(1,1))
2090 IF A<>&H55 THEN 2120
2100 A=ASC(INPUT$(1,1))
2110 IF A<>&H55 THEN A=&H1A
2120 VPOKE I,A
2130 A=ASC(INPUT$(1,1))
2140 IF A<>&H55 THEN 2170
2150 A=ASC(INPUT$(1,1))
2160 IF A<>&H55 THEN A=&H1A
2170 VPOKE I+&H2000,A
2180 NEXT
2190 CLOSE
2200 GOSUB 27000
2210 GOTO 21000
3000 REM --   カラー   --
3010 GOSUB 20000
3020 FOR I=1 TO 15
3030 LINE(24+I*8,176)-STEP(7,7),I,BF
3040 LINE-STEP(-7,-7),4,B
3050 NEXT
3060 P0=C:P1=8:P2=24:P3=176:P4=7:P5=7:P6=1:P7=15
3070 GOSUB 25000
3080 C=P0:COLOR C
3090 GOSUB 28000
3100 GOSUB 27000
3110 GOTO 21000
4000 REM --  ふとさ  --
4010 GOSUB 20000
4020 F=F MOD 3+1
4030 IF F=1 THEN MY=174:GOTO 4050
4040 MY=173
4050 GOSUB 28000
4060 GOTO 21000
5000 REM -- カーソル・スピード --
5010 GOSUB 20000
5020 CS=-1-CS
5030 GOSUB 28000
5040 GOTO 21000
6000 REM -- スティック/キー きりかえ --
6010 GOSUB 20000
6020 ST=1-ST
6030 GOSUB 28000
6040 GOTO 21000
```

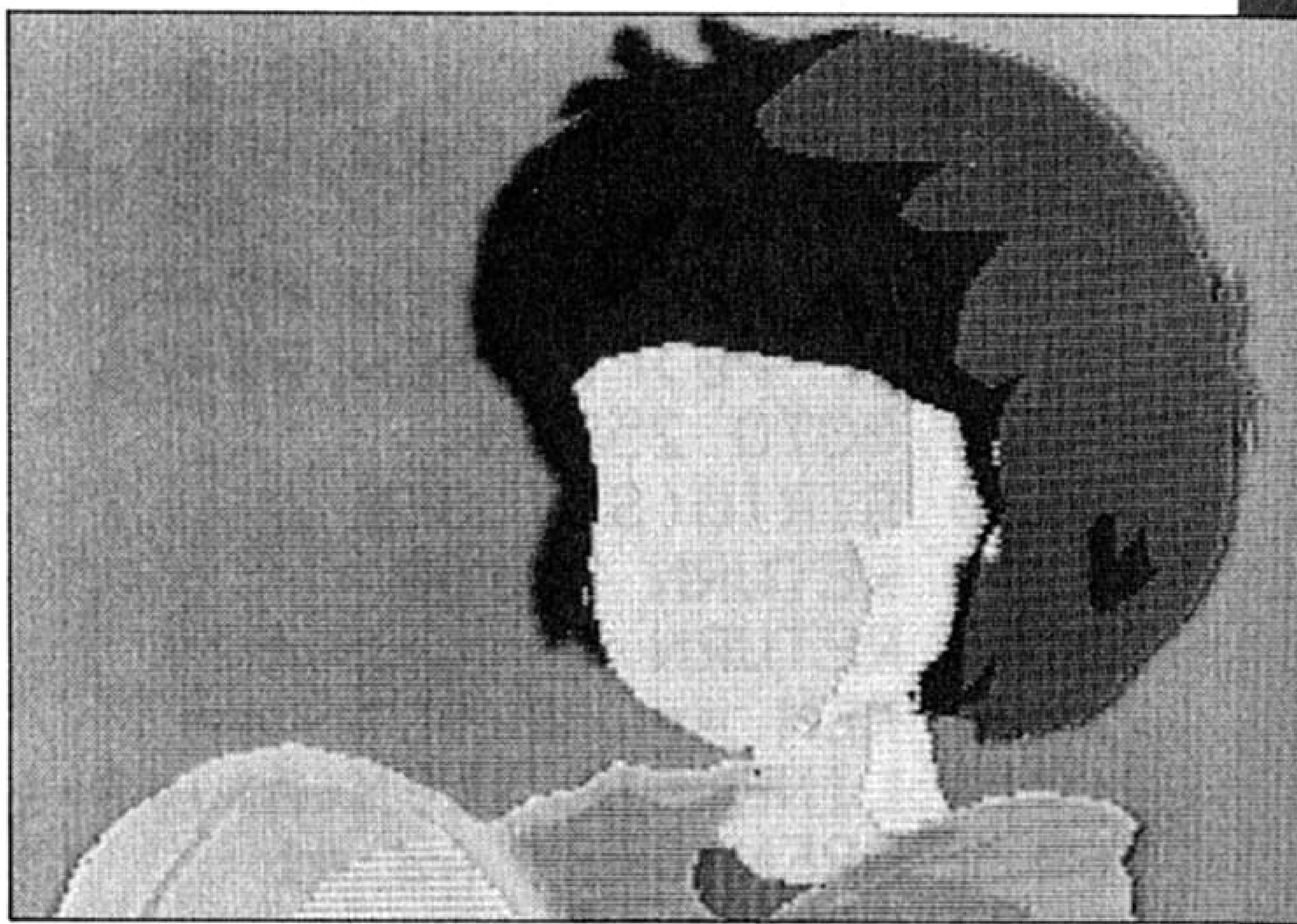


```
7000 REM --  セーブ  --
7010 GOSUB 20000
7020 DRAW"bm24,176"
7030 A$="セーブ ファイルめい "
7040 GOSUB 30000
7050 IF A$="" THEN 7190
7060 IF I$=CHR$(27) THEN 7190
7070 OPEN "CAS:"+A$ FOR OUTPUT AS #1
7080 FOR I=0 TO 5631
7090 A=VPEEK(I)
7100 IF A=&H55 THEN PRINT#1,"UU";:GOTO 7130
7110 IF A=&H1A THEN PRINT#1,"UV";:GOTO 7130
7120 PRINT#1,CHR$(A);
7130 A=VPEEK(I+&H2000)
7140 IF A=&H55 THEN PRINT#1,"UU";:GOTO 7170
7150 IF A=&H1A THEN PRINT#1,"UV";:GOTO 7170
7160 PRINT#1,CHR$(A);
7170 NEXT
7180 CLOSE
7190 GOSUB 27000
7200 GOTO 21000
8000 REM --  バック・カラー  --
8010 FOR I=1 TO 15
8020 LINE(24+I*8,176)-STEP(7,7),I,BF
8030 LINE-STEP(-7,-7),4,B
8040 NEXT
8050 P0=C:P1=8:P2=24:P3=176:P4=7:P5=7:P6=1:P7=15
8060 GOSUB 25000
8070 COLOR ,,P0
8080 GOSUB 27000
8090 GOTO 21000
9000 REM --  クリア  --
9010 GOSUB 20000
9020 A$="クリアー  する[リターン] しない[スペース]"
9030 DRAW"bm24,176"
9040 GOSUB 26000
9050 IF INKEY$<>"" THEN 9050
9060 A$=INPUT$(1)
9070 IF A$=" " THEN 9100
9080 IF A$<>CHR$(13) THEN 9060
9090 LINE(0,0)-(255,174),0,BF
9100 GOSUB 27000
9110 IF STRIG(ST) THEN 9110
9120 GOTO 21000
10000 REM --  ドット  --
10010 DRAW"bm=x;,=y;"
10020 ON F GOTO 10030,10050,10070
```



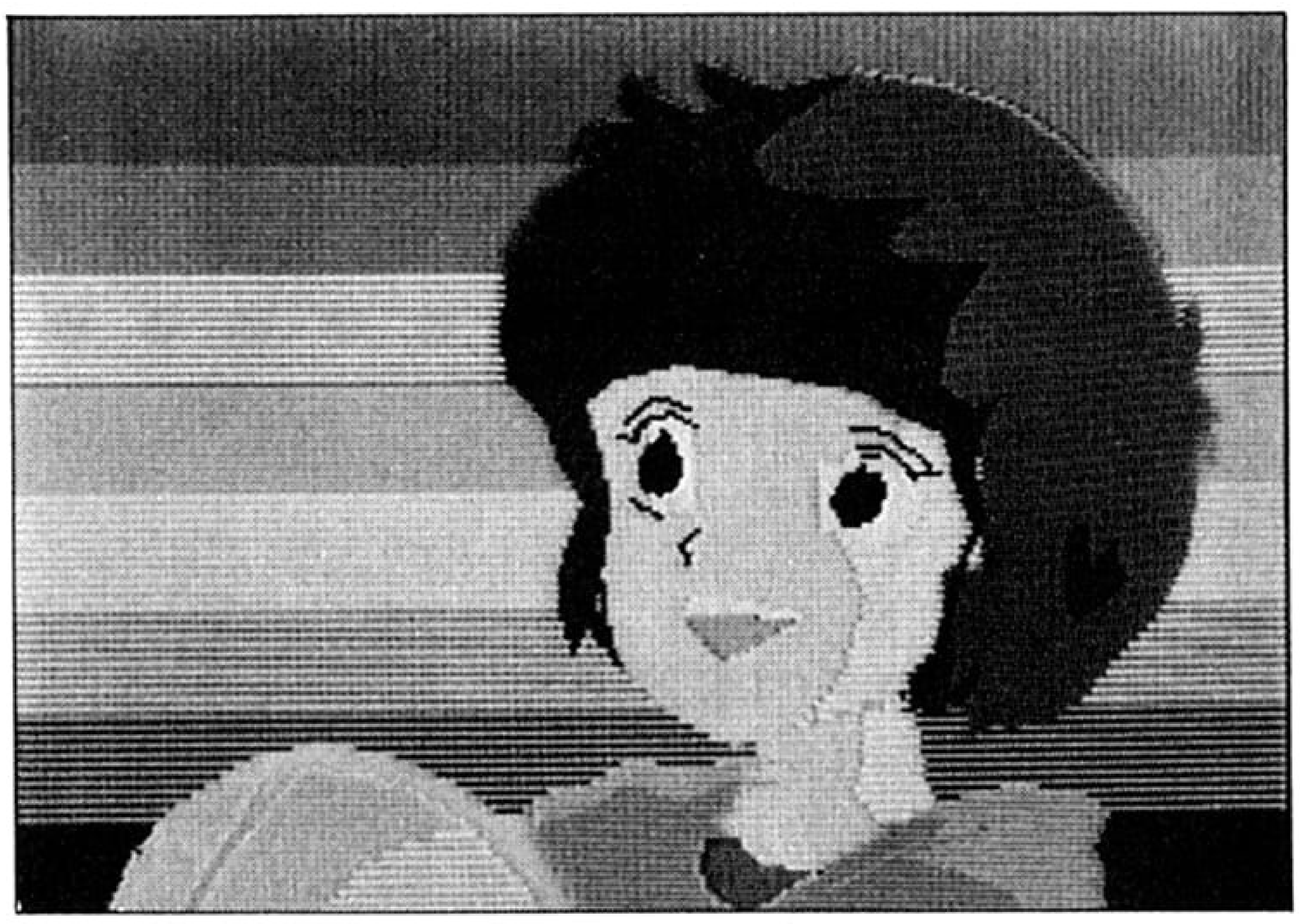
```
10030 PSET(X,Y)
10040 RETURN
10050 DRAW"c=c;s4d1l1u1r1"
10060 RETURN
10070 DRAW"c=c;s4l1d1r2u2l2d1"
10080 RETURN
11000 REM --  ライン  --
11010 IF FF THEN FF=0:GOTO 23000
11020 ON F GOTO 11090,11030,11050
11030 LINE(X1+1,Y1+1)-(X+1,Y+1)
11040 GOTO 11070
11050 LINE(X1,Y1-1)-(X,Y-1)
11060 LINE(X1-1,Y1)-(X-1,Y)
11070 LINE(X1,Y1+1)-(X,Y+1)
11080 LINE(X1+1,Y1)-(X+1,Y)
11090 LINE(X1,Y1)-(X,Y)
11100 FF=-1:GOTO 24000
12000 REM --  はこ  --
12010 IF FF THEN FF=0:GOTO 23000
12020 AX=X:AY=Y
12030 IF AX<X1 THEN SWAP AX,X1
12040 IF AY<Y1 THEN SWAP AY,Y1
12050 ON F GOTO 12110,12060,12090
12060 LINE(X1,Y1)-(AX+1,AY+1),,B
12070 LINE(X1+1,Y1+1)-(AX,AY),,B
12080 FF=-1:GOTO 24000
12090 LINE(X1+1,Y1-1)-(AX-1,AY+1),,B
12100 LINE(X1-1,Y1+1)-(AX+1,AY-1),,B
12110 LINE(X1,Y1)-(AX,AY),,B
12120 FF=-1:GOTO 24000
13000 REM -- はこ ぬりつぶし --
13010 IF FF THEN FF=0:GOTO 23000
13020 LINE(X1,Y1)-(X,Y),,BF
13030 FF=-1:GOTO 24000
14000 REM --  ペイント  --
14010 LINE(0,175)-(255,175)
14020 PAINT(X,Y),C
14030 LINE(0,175)-(255,175),4
14040 RETURN
15000 REM --  サークル  --
15010 IF FF THEN FF=0:GOTO 23000
15020 R=SQR((X-X1)*(X-X1)+(Y-Y1)*(Y-Y1))
15030 IF R=0 THEN 10000
15040 R#=R*R-(176-Y1)*(176-Y1)
15050 IF R#<0 THEN 15170
15060 R#=SQR(R#)/R
15070 R#=ATN(R#/SQR(-R#*R#+1))
```

```
15080 ON F GOTO 15150,15090,15110
15090 CIRCLE(X1+1,Y1+1),R,,4.71238898#+R#,4.71238898#-R#
15100 GOTO 15130
15110 CIRCLE(X1,Y1-1),R,,4.71238898#+R#,4.71238898#-R#
15120 CIRCLE(X1-1,Y1),R,,4.71238898#+R#,4.71238898#-R#
15130 CIRCLE(X1,Y1+1),R,,4.71238898#+R#,4.71238898#-R#
15140 CIRCLE(X1+1,Y1),R,,4.71238898#+R#,4.71238898#-R#
15150 CIRCLE(X1,Y1),R,,4.71238898#+R#,4.71238898#-R#
15160 FF=-1:GOTO 24000
15170 ON F GOTO 15240,15180,15200
15180 CIRCLE(X1+1,Y1+1),R
15190 GOTO 15220
15200 CIRCLE(X1,Y1-1),R
15210 CIRCLE(X1-1,Y1),R
15220 CIRCLE(X1,Y1+1),R
15230 CIRCLE(X1+1,Y1),R
15240 CIRCLE(X1,Y1),R
15250 FF=-1:GOTO 24000
20000 REM ==  ｷｰ ｵﾌ  ==
20010 STRIG(0) OFF
20020 STRIG(1) OFF
20030 FOR I=1 TO 9
20040 KEY (I) OFF
20050 NEXT
20060 RETURN
21000 REM ==  ｷｰ ｵﾝ  ==
21010 FOR I=1 TO 9
21020 KEY (I) ON
21030 NEXT
21040 RETURN
22000 REM == ｶｰｿﾙいどう ==
22010 ON STRIG GOSUB 22140,22140
22020 STRIG(ST) STOP
22030 X=X+DX(STICK(ST))
22040 Y=Y+DY(STICK(ST))
22050 IF X>MX THEN X=NX
22060 IF X<NX THEN X=MX
22070 IF Y>MY THEN Y=NY
22080 IF Y<NY THEN Y=MY
22090 PUT SPRITE 0,(X-3,Y-4),CU
22100 STRIG(ST) ON
22110 IF CS THEN 22020
22120 FOR I=0 TO 99:NEXT
22130 GOTO 22020
22140 STRIG(ST) OFF
22150 RETURN 22160
22160 RETURN
```



```
23000 REM == サブカーソル オン ==
23010 PUT SPRITE 1,(X-3,Y-4),C
23020 X1=X:Y1=Y
23030 RETURN
24000 REM == サブカーソル オフ ==
24010 PUT SPRITE 1,(0,208)
24020 RETURN
25000 REM == カーソルいどう(2) ==
25010 ON STRIG GOSUB 25130,25130
25020 LINE(P0*P1+P2,P3)-STEP(P4,P5),15,B
25030 STRIG(ST) ON
25040 FOR I=0 TO 99:NEXT
25050 STRIG(ST) STOP
25060 B=DX(STICK(ST))
25070 IF B=0 THEN 25030
25080 LINE-STEP(-P4,-P5),4,B
25090 P0=P0+B
25100 IF P0<P6 THEN P0=P7
25110 IF P0>P7 THEN P0=P6
25120 GOTO 25020
25130 STRIG(ST) OFF
25140 RETURN 25150
25150 RETURN
26000 REM ==  ひょうじ  ==
26010 LINE STEP(LEN(A$)*8-1,7)-STEP(-LEN(A$)*8+1,-7),4,BF
26020 OPEN"GRP:"AS #1
26030 COLOR 15
26040 PRINT#1,A$;
26050 COLOR C
26060 CLOSE
26070 RETURN
27000 REM == メッセージ クリア ==
27010 LINE(0,175)-(255,183),4,BF
27020 RETURN
28000 REM ==  じょうほう  ==
28010 A$=MID$("DOT   LINE  BOX   FILL  PAINT CIRCLE",M
*6-5,6)
28020 DRAW"bm64,184"
28030 GOSUB 26000
28040 A$=CHR$(48+F)
28050 DRAW"bm160,184"
28060 GOSUB 26000
28070 LINE(176,185)-STEP(16,6),C,BF
28080 IF CS THEN A$="FAST":GOTO 28100
28090 A$="SLOW"
28100 DRAW"bm200,184"
```

```
28110 GOSUB 26000
28120 IF ST THEN A$=" J":GOTO 28140
28130 A$=" K"
28140 GOTO 26000
29000 REM ==  カーソルてんめつ  ==
29010 CU=16-CU
29020 INTERVAL ON
29030 RETURN
30000 REM ==  にゅうりょく  ==
30010 GOSUB 26000
30020 OPEN"GRP:" AS #1
30030 COLOR 15
30040 IF INKEY$<>"" THEN 30040
30050 A$=""
30060 PRINT#1,CHR$(95);
30070 I$=INPUT$(1)
30080 IF I$=CHR$(13) OR I$=CHR$(27) THEN CLOSE:COLOR C
:RETURN
30090 IF I$=CHR$(8) THEN 30150
30100 IF LEN(A$)>10 THEN 30070
30110 IF I$<" " THEN 30070
30120 A$=A$+I$
30130 LINE STEP(-1,7)-STEP(-7,-7),4,BF
30140 PRINT#1,I$;:GOTO 30060
30150 IF LEN(A$)=0 THEN 30070
30160 LINE STEP(-1,7)-STEP(-15,-7),4,BF
30170 A$=LEFT$(A$,LEN(A$)-1)
30180 GOTO 30060
```



FOUR

Part **4**

Let's get C.G.

色バケ現象・その傾向と対策

MSXのグラフィックカラーが全部で16色あることは，すでに何度も説明した．絵を描くためには多くの色が出せることがもちろん大切なことで，その点ではとてもありがたい．けれどもせっかくの16色機能も，画面モードが2のときは，色の使い方に「横に8ドットを一組として，その中では2色しか使ってはいけない」という制限がある．これがMSXで絵を描きたいという人にとってはとてもやっかいな代物(しろもの)で，色を付けていく途中で色々な（別にシャレのつもりではない）問題を引き起こしてしまう．

そこで本節では，この制限がなぜ起きるのかという点や，制限をかいくぐる方法などを中心に話を進めていこう．

4.1 赤が緑に変わるとき

まず，実際に横8ドットの中でむりやり3色を使おうとした場合に，いったいどうなってしまうのかを見てみよう．次のプログラムを打ちこんでほしい．

```
10 SCREEN 2
20 COLOR 15,14,7:CLS
30 LINE(121,0)-(121,191),9
40 LINE(122,0)-(122,191),9
50 D$=INPUT$(1)
60 LINE(124,0)-(124,191),2
70 GOTO 70
```

簡単なプログラムだから，すでにここまで本書を読み進んできた人には，どんな作業を行っているのか見当がつくと思う．

まず，灰色の画面に縦に赤い線を引き（2本引いたのは，太い線にして見やすくするためだ），次に何かのキーを押すとその右側に緑の線を1本引くというプログラムである．

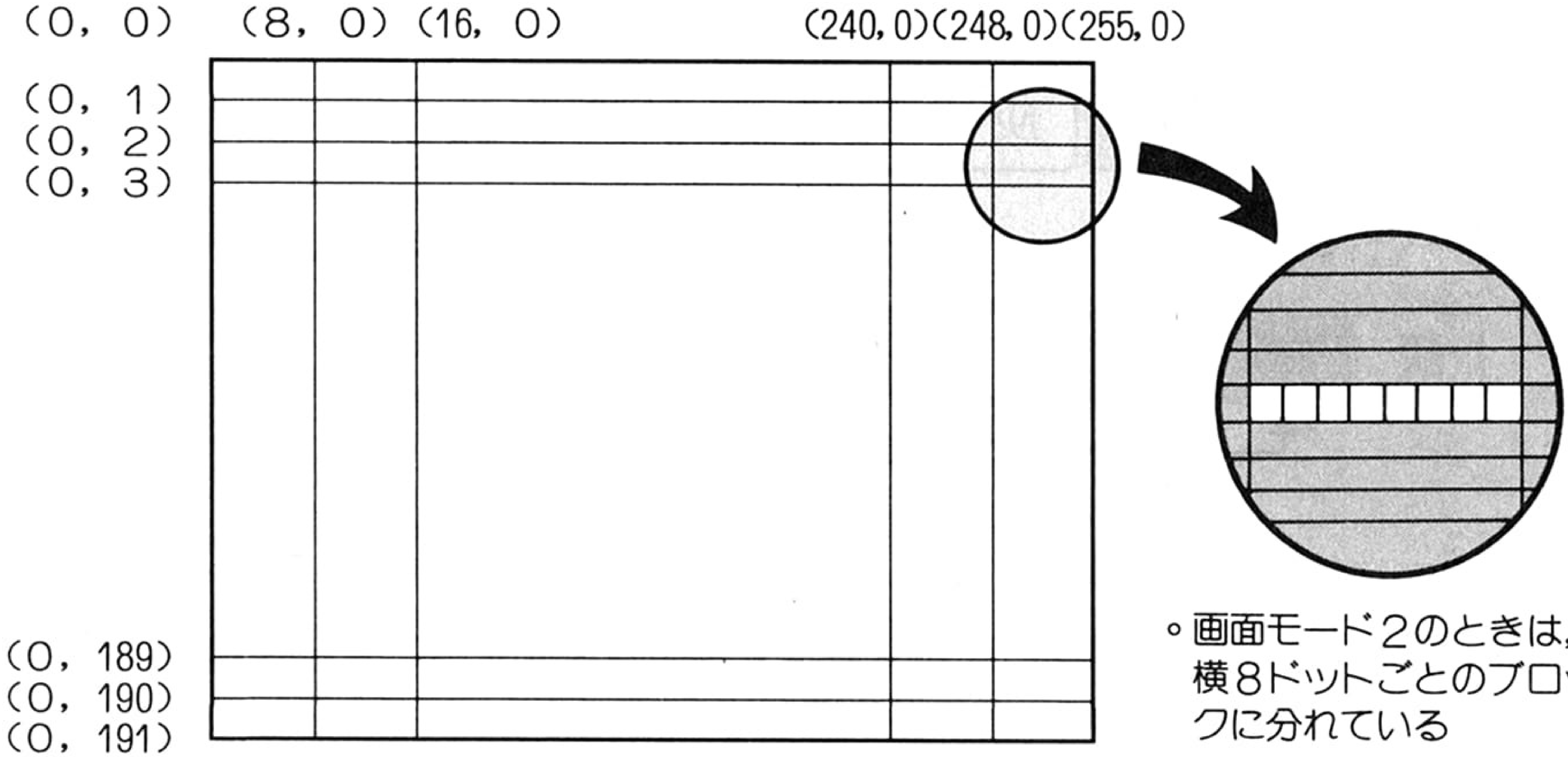
打ちこみが終わったら，さっそくRUNしてみよう．画面が灰色でクリアされると，中央あたりに赤い縦線が表示される．ここまではいつもと同じだ．

次にどれかキーを押してみよう．すると，右側に緑の線が出る．これも当然のことだ——が，なんと，よく見ると，もともと描かれていた赤の線が，いつのまにか緑色に変わってしまっている．

けれども，これは読者がお持ちのMSXマシンが故障したわけでも，プログラムがまちがっていたわけでもない．これが横8ドットの一組の中に，3色目を使用したために起きた，「色バケ現象」なのである．

4.2 定員2名の8畳間

MSXパソコンは，画面のための情報をおさめておく記憶装置（これをV－RAMという）があまり大きくない．そのくせ16色も出せる理由は，画面の色に関する情報を，8ドットごとひとまとめにして決めているからだ．

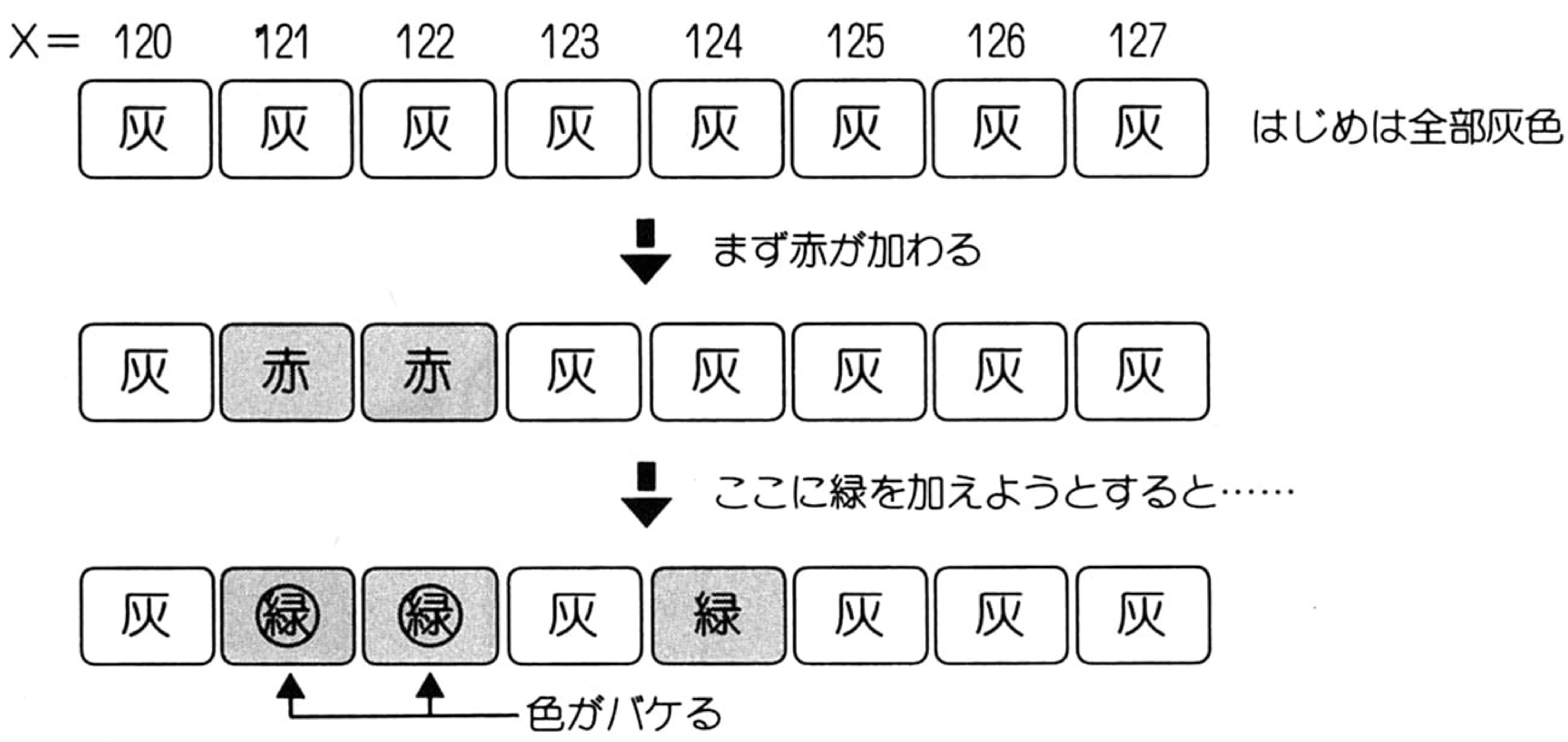


◦画面モード2のときは，横8ドットごとのブロックに分れている

前ページの図のように，MSX の画面は横に 8 ドットをひとまとまりとして，ヨコ32 個×タテ 192 個のブロックに分かれている．そして，それぞれのブロックごとに，色に関する情報をおさめておく記憶場所が用意されている．実は，「横 8 ドットにつき 2 色まで」という制限は，この色情報を格納する記憶場所が，2 色分の広さしかないために起きるのだ．

さきほどのプログラムの例にもどって考えてみよう．この場合には，ちょうど横方向のX座標が120～127の位置にある各ブロックに対して，2 色のちがう色の直線を描きこもうとしていることがわかる．

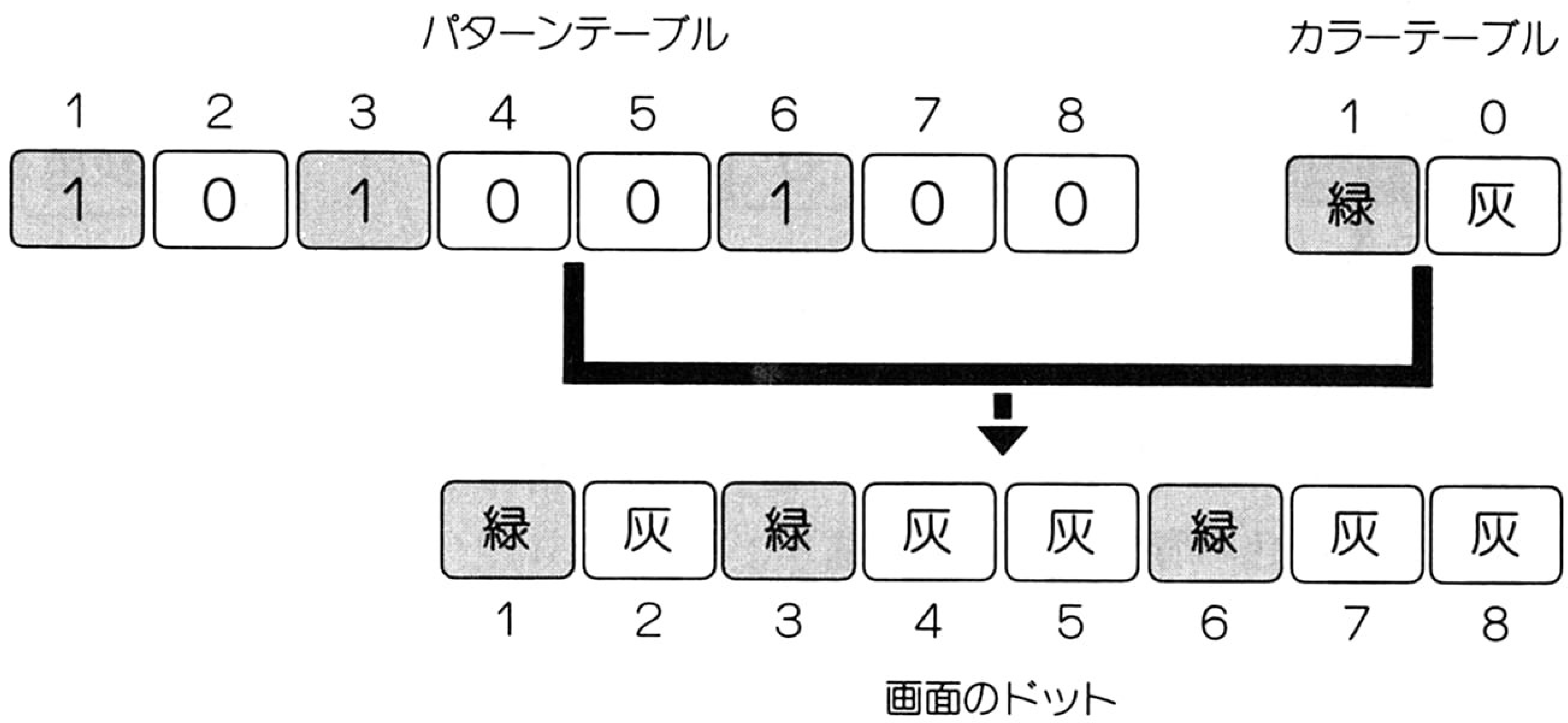
プログラムの流れを順に追ってみると，まず10行目と20行目を実行した時点で，このブロック内の 8 ドットはすべて灰色になっている．次に30行と40行の LINE 命令で，左から 2 番目と 3 番目のドットが赤に変わる．50行は単なるキー入力待ちだが，何かキーが押されて60行目へ進むと，3 色目として 5 番目のドットを緑に変えるようにとの指示が書かれている．このとき，色の情報をおさめる場所は 2 色分しか用意されていないので，MSX はこの 8 ドットのブロックが 2 色におさまるように赤の部分もチェンジしてしまうわけだ．



ではこの「色バケ」現象は，どんな命令を指示したときに，どんな色に変わってしまうのだろうか． このへんの因果関係を知るためには，もう少しくわしく，MSXの記憶装置の内部に立ち入ってみる必要がある．

4.3 テーブルの下のやりとり

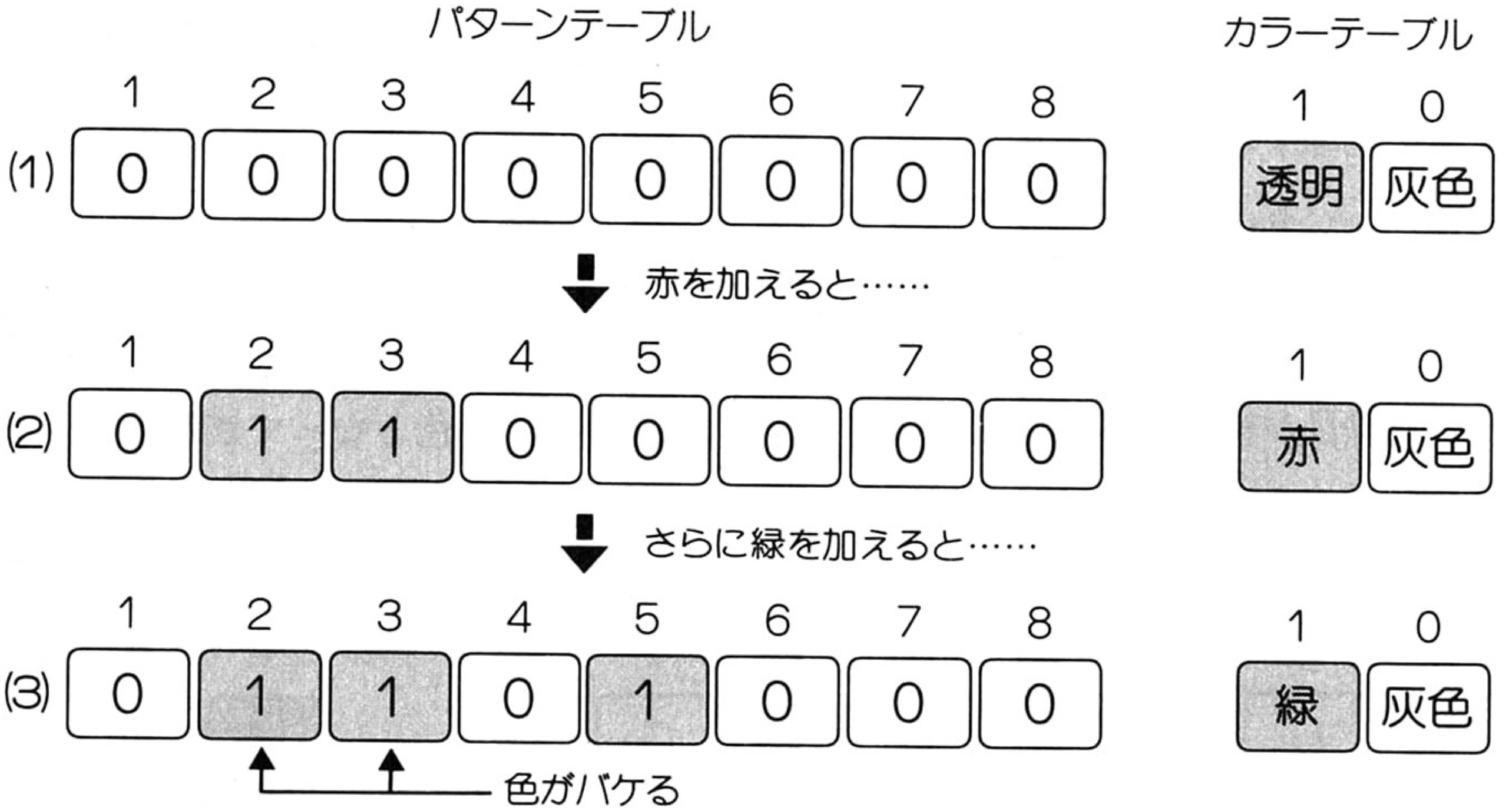
MSX の内部には，8 ドット＝1 ブロック分の画面情報を格納するために，各ブロックごとに「パターンテーブル」と「カラーテーブル」の 2 つが用意されている．このうち，まず「パターンテーブル」というのは，ブロック内の各ドットの状態を記憶するためのもので，下の図のように「1」か「0」のどちらかにセットされている．これに対して「カラーテーブル」のほうは，パターンテーブル上の「1」が何色で，「0」が何色であるかを記憶する 2 つの部分に分かれている．たとえば，2 つのテーブルが図のような状態になっているときには，ドットの 1，3，6 番目が緑色，残りの 2，4，5，7，8 番目が灰色となって表示される仕組みだ．



ここでもう一度，前にあげたプログラムについて，パターンテーブルとカラーテーブルの変化の様子を追いかけてみよう．再び同じリストをのせておく．

```
10 SCREEN 2
20 COLOR 15,14,7:CLS
30 LINE(121,0)-(121,191),9
40 LINE(122,0)-(122,191),9
50 D$=INPUT$(1)
60 LINE(124,0)-(124,191),2
70 GOTO 70
```

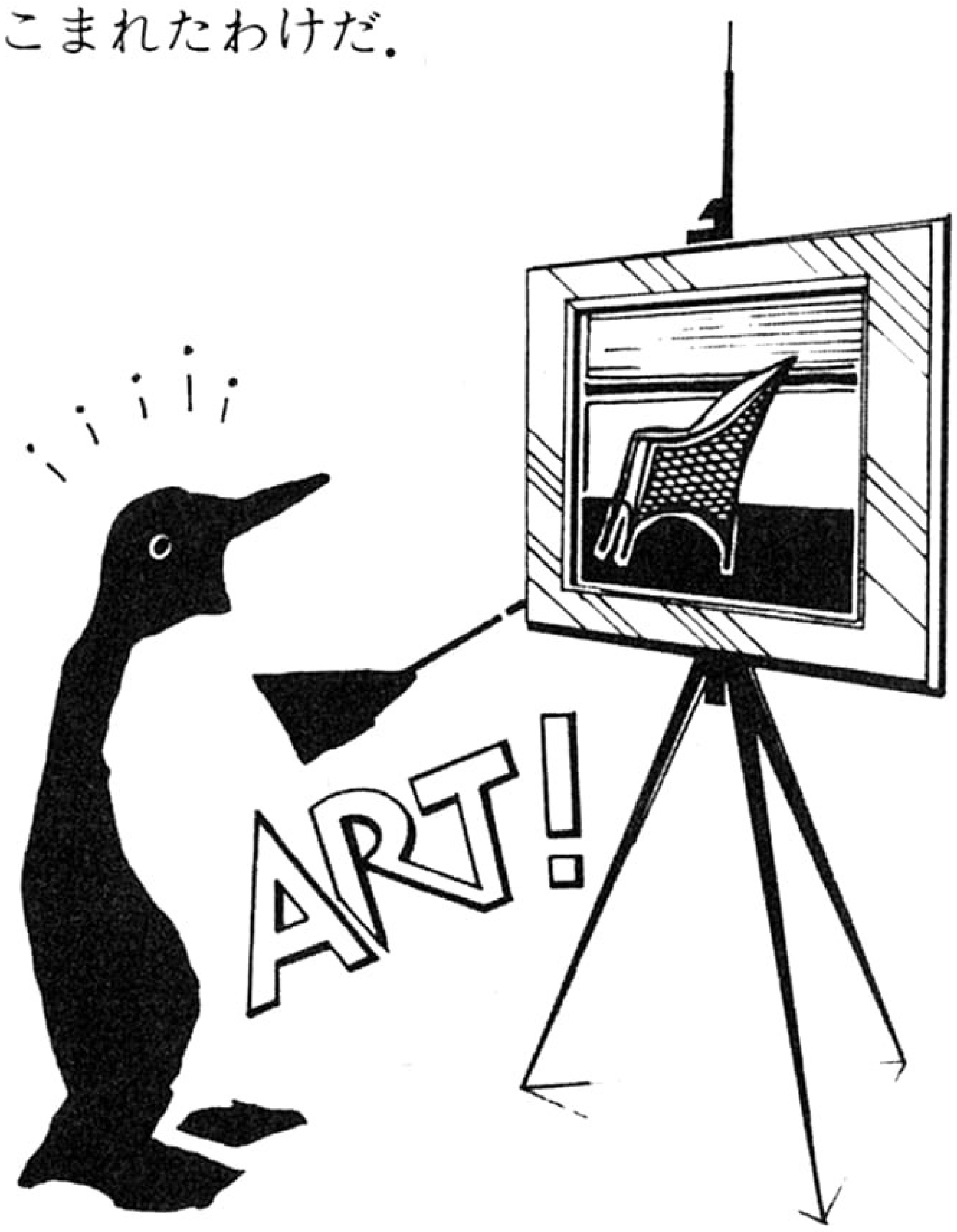
まず10行目と20行目．下の図の(１)のように，パターンテーブルはすべて「０」にセットされ，カラーテーブルの「０」を表わす部分には「灰色」のコードが記憶されている．



次に30行目と40行目が実行されると，パターンテーブルの２番と３番が「１」になり，カラーテーブルの「１」には「赤」という情報が書きこまれる〔図の（２)〕．これで灰色の背景に赤い線が描きこまれたわけだ．

ところが，60行目で５番目のドットを緑にしなさいと命令されたMSXは，さあ大変，もうすでにカラーテーブルはいっぱいになっているし，どうしようもない．やむなくMSXはパターンテーブルの５番を「１」とした上で，カラーテーブルの「１」の情報を赤から緑に変えてしまった〔図の（３)〕．

こうして画面上ではこれまで赤だったドットの２番と３番までもが，緑色に化けてしまうわけだ．



4.4 オール1なら0から出発

もう少し例をあげて考えてみよう．左ページの図の（3）の状態から，さらに4，6，7，8番目のドットを黒にかえるよう命令してみる．すると，下の図の（4）のように，パターンテーブルは1番目だけが「0」で，残りはすべて「1」になる．カラーテーブルのほうは「1」の色として黒，「0」の色として灰色が書きこまれる．ブロック内は先頭の1ドットを除いてすべて黒に化けてしまうわけだ．

ではここで，1番目のドットを赤にしようとすると，どうなるだろうか．実はこの場合は，MSX はパターンテーブルのほうには手を触れず，カラーテーブルの「0」のところを灰色から赤に変えて，すました顔をしている〔図の（5）〕．

パターンテーブル　カラーテーブル

	1	2	3	4	4	6	7	8	1	0
(3)	0	1	1	0	1	0	0	0	緑	灰色

↓ 4,6,7,8番を黒にかえると……

	1	2	3	4	5	6	7	8	1	0
(4)	0	1	1	1	1	1	1	1	黒	灰色

↓ 1番を赤にすると……

	1	2	3	4	5	6	7	8	1	0
(5)	0	1	1	1	1	1	1	1	黒	赤

こちらは変化しない　ここが赤にかわる

	1	2	3	4	5	6	7	8	1	0
(6)	1	1	1	1	1	1	1	1	黒	赤

↓ たとえば2番を緑にかえると……

	1	2	3	4	5	6	7	8	1	0
	1	0	1	1	1	1	1	1	黒	緑

↑ここを0にする　ここを緑にする

これは，図の（6）のようにパターンテーブルがすべて「1」で埋まっている場合も同じで，たとえばこの状態からどれか1つのドットを緑色に変えようとすると，MSXはパターンテーブルの対応する部分を「0」に書きかえ，同時にカラーテーブルの「0」の色を緑に変えて応じてくる．

さて，ここまでにあげた例を見てくると，一見気まぐれに思えるMSXの色バケ現象も，きちんと一定の法則にのっとって色が変化させられていることに気づく．この法則をよく調べてみると，新たな色で点を打とうとする場合には，

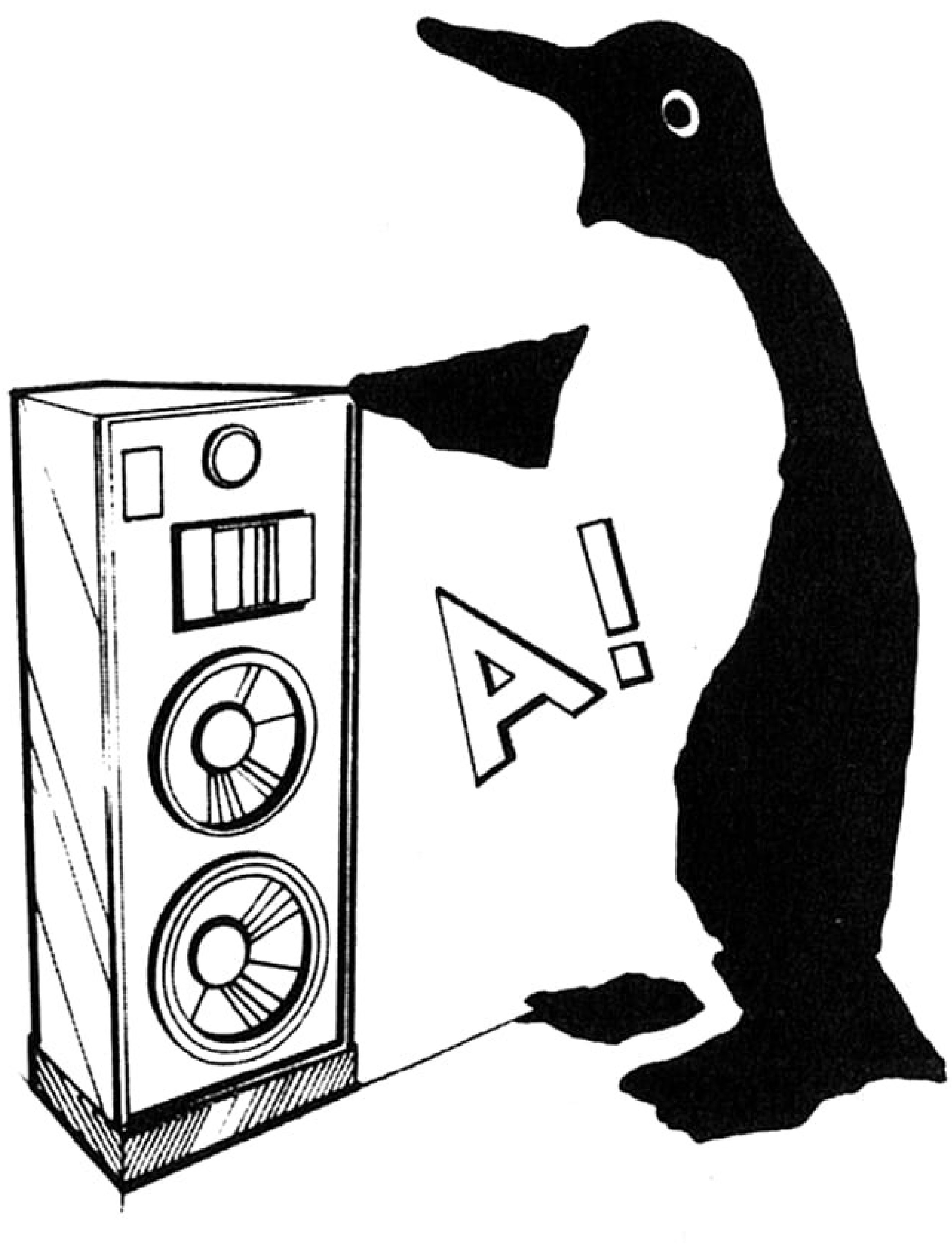
ひとつのブロック内で，他のドットのパターンはすべて「1」か？

- **「はい」なら，そのドットを「0」にして，カラーテーブルの「0」の色を指定された色にする．**
- **「いいえ」なら，そのドットを「1」にして，カラーテーブルの「1」の色を指定された色にする．**

という動作を行っていることがわかる．

ところで，この動作には例外もある．PAINT命令や，LINE・BF命令を使って色を塗りつぶす場合に，あるブロックのパターンテーブルのデータがすべて「1」であったときには，これを全部「0」に書きかえた上で，カラーテーブルの「0」の方を指定された色に書き直す．

たとえば，右上の2つの命令をそれぞれ実行すると，結果は図のようになり，画面の上での見かけはまったく変わらないが，MSXの内部ではちょうどパターンとカラーの情報が正反対の形になる．これは，パターンテーブルの中味として，できるだけ「0」を多くしておくことで，新たな色バケが起こる範囲がひろがらないように配慮を加えてあるわけだ．



(1)
```
LINE(0,0)-(7,0),2
```

(2)
```
LINE(0,0)-(7,0),2,BF
```

(1) LINEの場合

X=	0	1	2	3	4	5	6	7		1	0
	1	1	1	1	1	1	1	1		緑	

(2) LINE･BFの場合

X=	0	1	2	3	4	5	6	7		1	0
	0	0	0	0	0	0	0	0			緑

以上，MSXの色塗りの方法をまとめると，できるかぎり色バケが起きないように努力して，どうしてもダメなときには，カラーテーブルの「1」に対応する色を新しい色に変えてしまうということになる．この画面情報についての知識さえあたまに入れておけば，かなりこみ入った図形に色をつける場合でも，制限をかいくぐってうまく表示することが可能なわけだ．

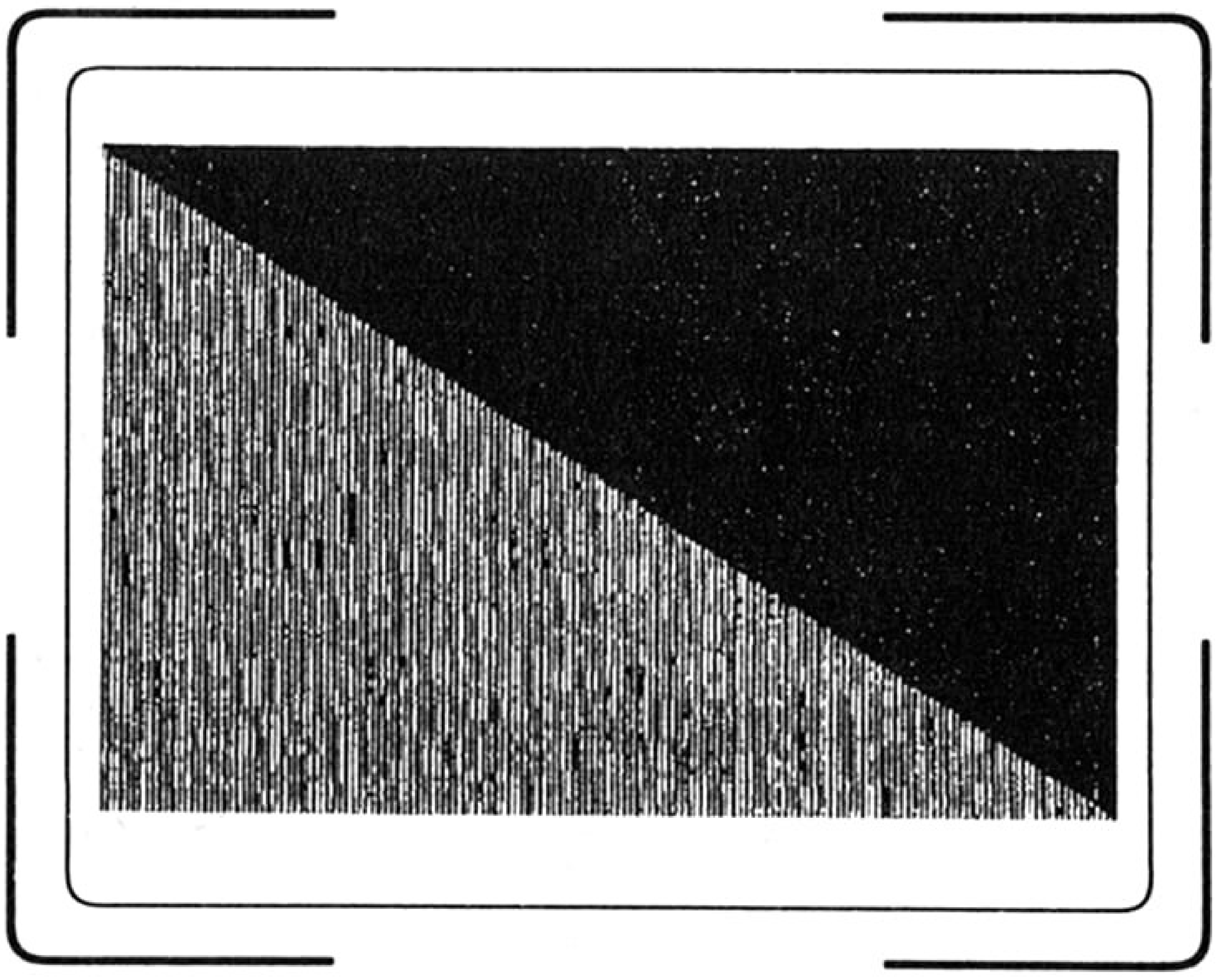
4.5 色バケの予防と治療

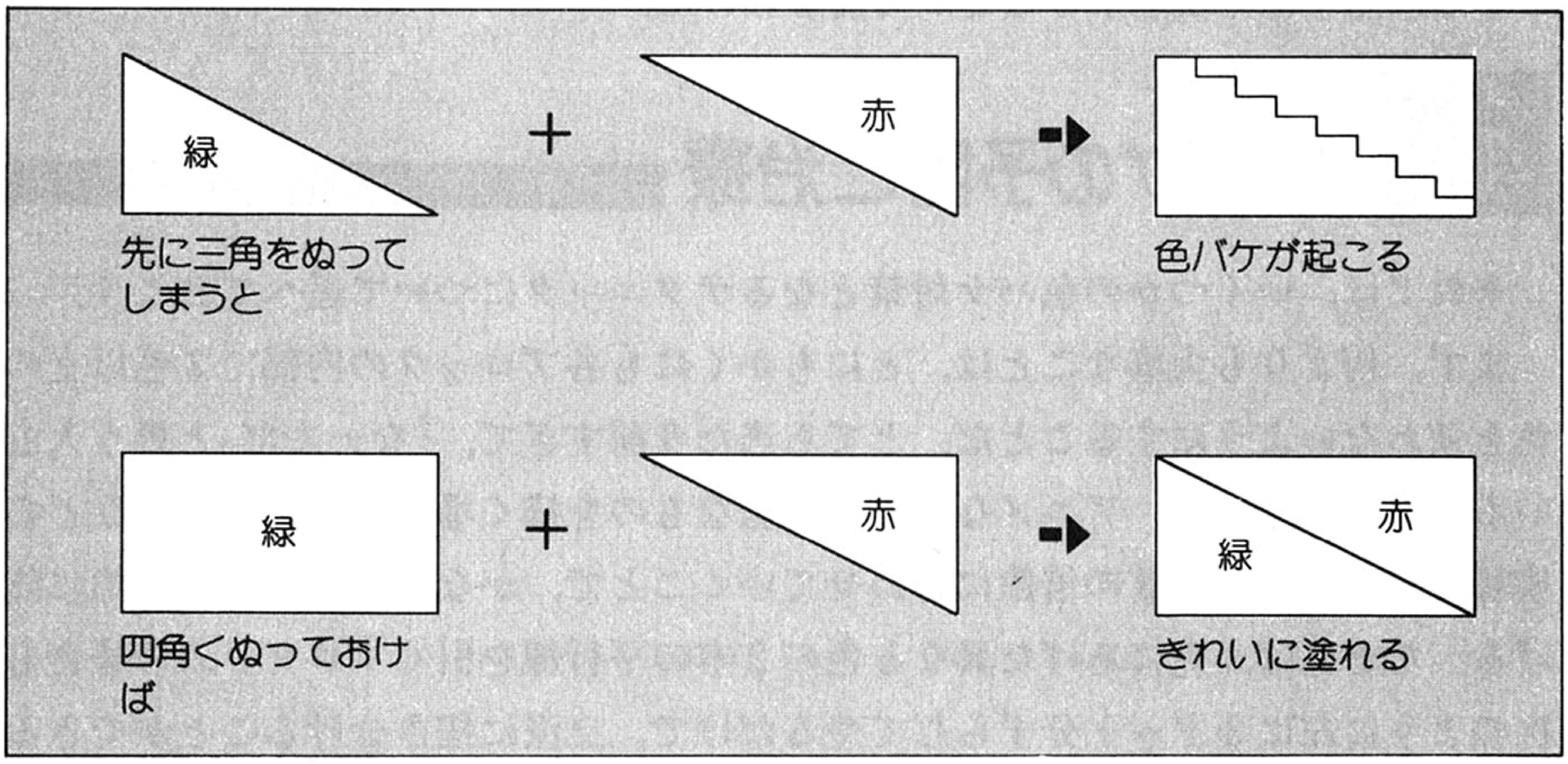
それでは，いくつかの色バケ対策となるテクニックについて述べておこう．

まず，何よりも大事なことは，とにもかくにも各ブロックの内部で3色以上の色を使わないようにすることだ．とても当たり前すぎて，「なーんだ」と思う人もいるかもしれないが，アニメなどの不定型なものを描く場合にも，顔や目などの横幅をできるかぎり8の倍数に合わせていくことで，かなりの色バケが未然に防げる．たとえば，先にあげた異なる色の2本の平行線を引くプログラムの場合も，次のように左に3ドット分ずらしてやるだけで，立派に描き分けることができる．

```
10 SCREEN 2
20 COLOR 15,14,7:CLS
30 LINE(118,0)-(118,191),9
40 LINE(119,0)-(119,191),9
50 D$=INPUT$(1)
60 LINE(121,0)-(121,191),2
70 GOTO 70
```

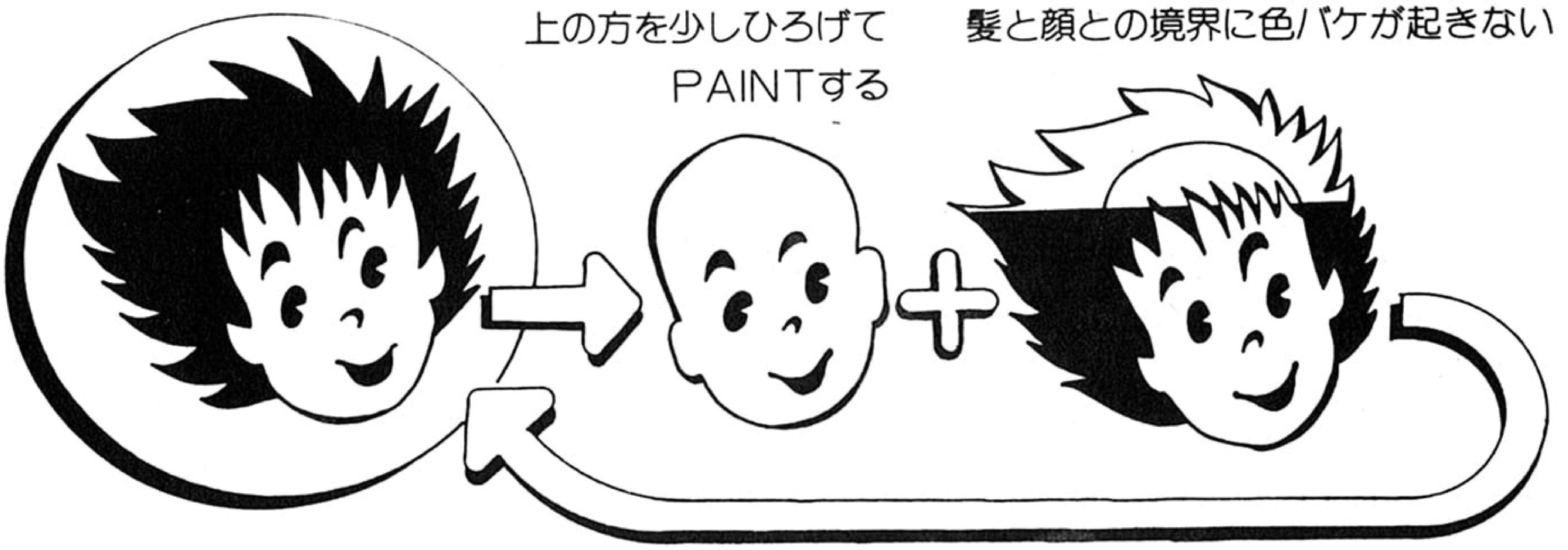
次に, PAINTの手順を変えることで, 色バケを防ぐ例をあげておこう. 下の図のように, 画面を対角線で2つに区切り, 別々の色で塗り分ける場合を考える.





このとき，手順として図の（1）のように，まず緑で三角に囲ってPAINTをかけたのち，次は赤でPAINTしようとすると，境界線付近で派手な色バケが起こってしまう(緑と赤と，背景の色の3色を使う部分が一時的にできるため)．そこで，こういうときは（2）のように，まず画面全体を緑で塗りつぶしておいてから，対角線を引き，赤でPAINTすれば問題はない．

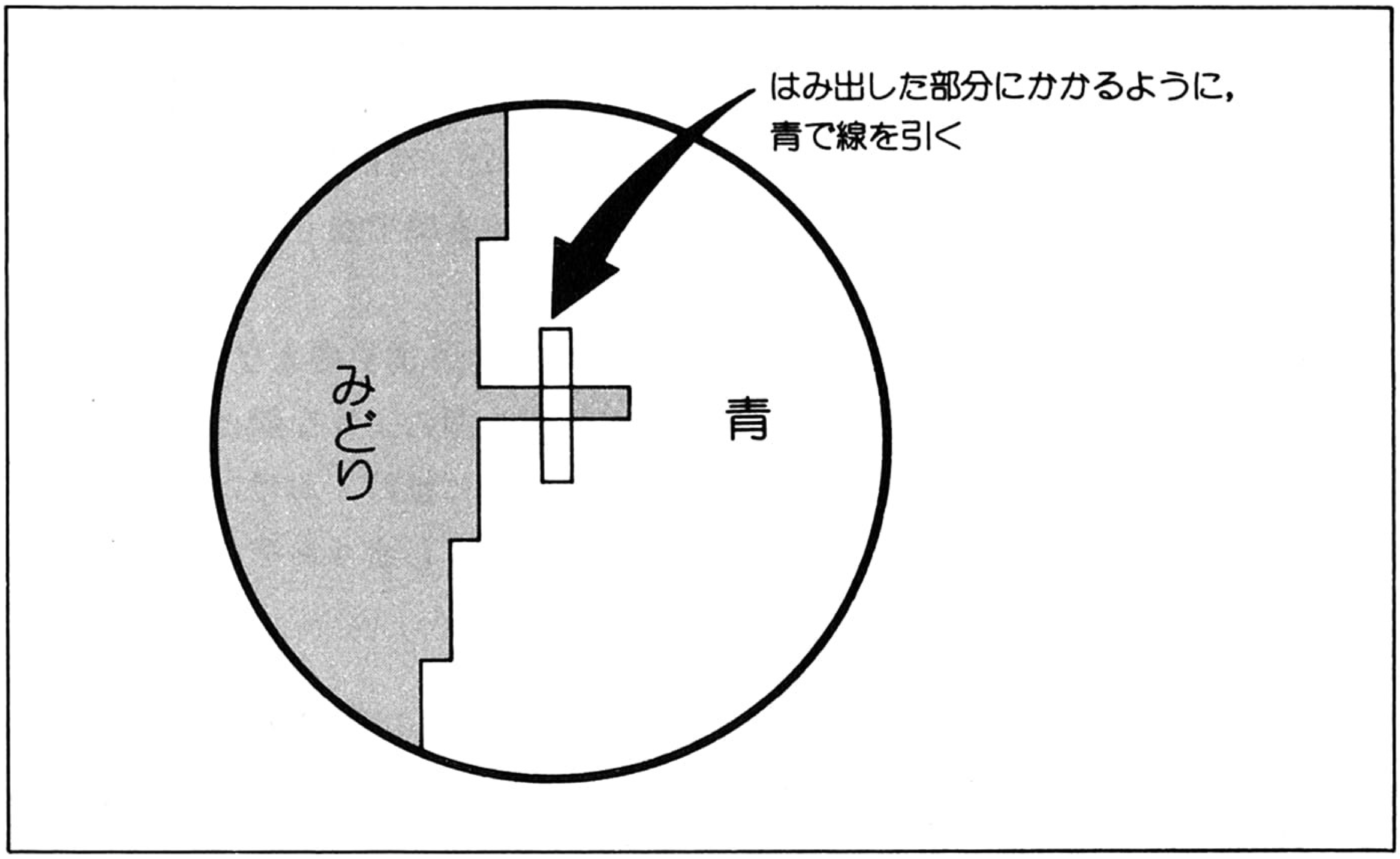
実際に人の顔などを描くときにも，PAINTの順序をうまく考えないと，色バケが起きてしまうことがたびたびある．たとえば，下の図のような顔と髪型を描くときには，あらかじめ顔の部分のPAINTの範囲を上に拡げておけば，あとで髪形の境界線を引くときにも1ブロックに3色が重なってしまうおそれがなくなって，色バケを防げる．



どうしても色がはみ出してしまうときには，はみ出したところをあとで修正するという手もある．次ページの図のように，緑が青の部分に飛び出したら，上から縦に青で線を引いてやると，色バケが目立たなくなる．

次のリストは，この原理を目で確かめるためのものだ．

```
10 SCREEN 2
20 COLOR 15,10,7:CLS
30 LINE(123,50)-(199,120),4,BF
40 LINE(82,90)-(152,20),3
50 A$=INPUT$(1)
60 LINE(124,50)-(124,60),4
70 GOTO 70
```



実行すると，はじめに描いた青い四角形の角が，明るい緑に変わってしまう．このあと何かキーを押してやると，色バケを起こした部分がもとにもどるはずだ．

これらのテクニックは，巻末にまとめて載せたアニメ・キャラクタを描くプログラムの中でも十分に駆使されているので，絵が描かれていく手順をよく覚えて，参考にしてほしい．ツール・プログラムを使ってオリジナルの絵を入力していく際にもかならず役に立つはずである．

4.6 もう一つの「色バケ」現象

最後にもう一つ，これまで説明してきたのとは別の理由から，色が変化してしまう現象について述べておこう．

プログラムのリストなどを画面に表示させたときに，背景の色が黒に設定されていると，よく見ると白で書いたはずの文字が，赤・青・緑の3色ににじんでいることに気づく．これは，MSX が映像出力の規格として採用している NTSC 信号方式に特有の現象で，家庭用テレビをディスプレイとして使う場合の宿命ともいえるものだ．この「色のにじみ」は，当然，LINE 命令などで線を引く場合にも影響してくる．

```
10 SCREEN 2
20 COLOR 15,1,7:CLS
30 LINE(120,0)-(120,191),15
40 LINE(127,0)-(127,191),15
50 LINE(134,0)-(134,191),15
60 LINE(142,0)-(147,191),15
70 GOTO 70
```

上のリストは，黒い背景に対して白の線を斜めに一本，縦に３本引くというものだ．ところがこれを打ちこんで実行してみると，ディスプレイによってはなぜか４本とも白には見えない．縦の線は，赤，青，緑の３色ににじんで見えるし，斜めの線にいたっては，きれいに３色に塗りわけられて見える始末だ．

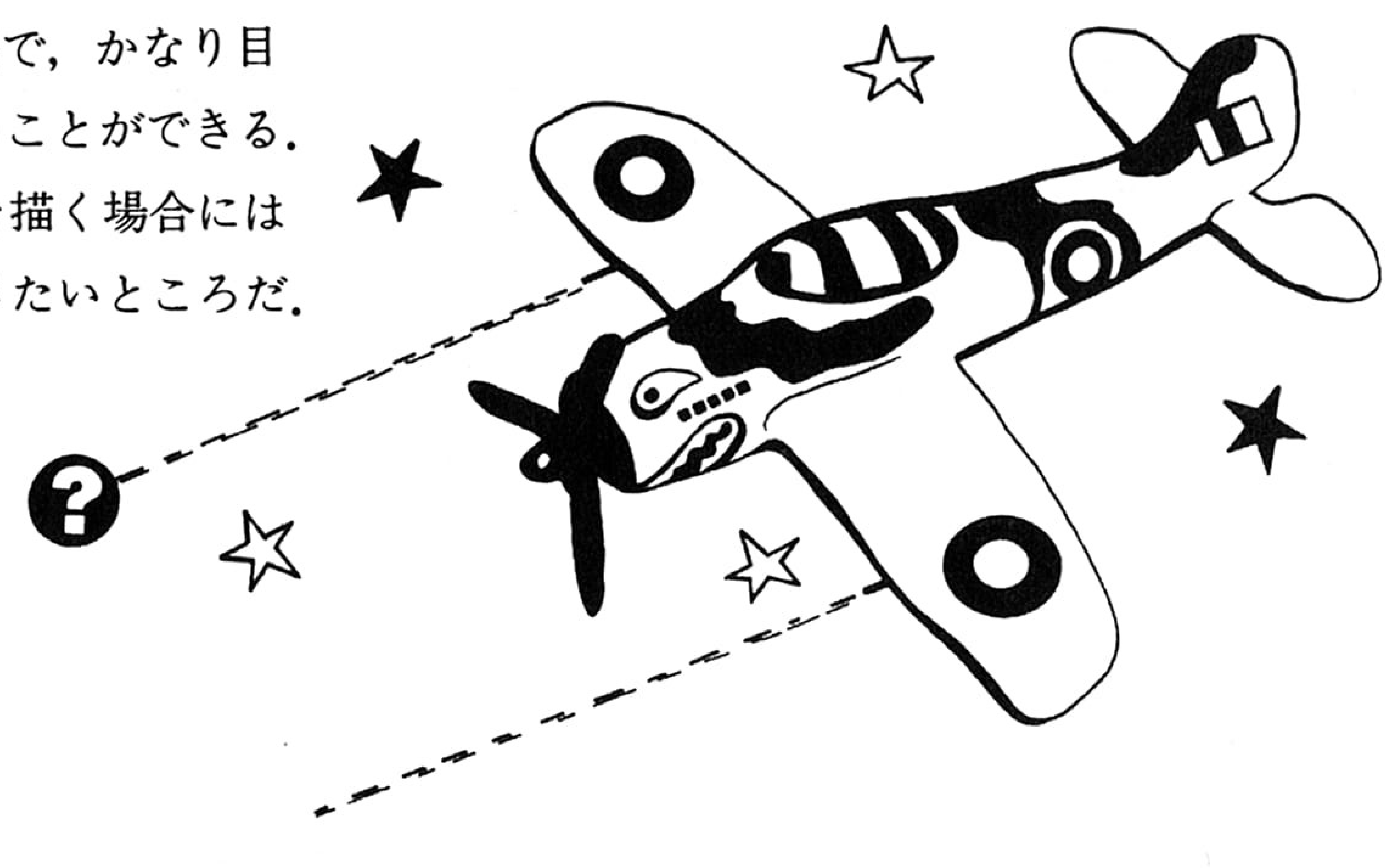
この例などは，わざと一番極端な場合をとり上げたわけだが，実際に画面に絵を描いていく上でも，この色のにじみは気になることが多い．

色のにじみは，先にも述べたように，ディスプレイの信号方式についてまわる現象なので，これを防ぐ手立ては正直いってほとんどない．けれども，

①背景を黒くすると，色のにじみが目立つので，なるべく使わないようにする．
②単独で直線を一本引くことを避けて，２本以上を合わせて太くするか，PAINT で塗りつぶして面として使う．

などの方法で，かなり目立たなくすることができる．

細かい絵を描く場合には注意しておきたいところだ．



アニメ・キャラクタ サンプルデータ集

次に載せる2つのリストは，いずれもアニメ用ツール「GT-2」を使って描いた絵を，[F･3]キーの機能によりBASICのプログラムとして生成したものだ．100行〜400行が絵を再現するルーチン，1,000行以下がデータ部分となっている．参考にしてほしい．

〈サンプルデータ 1〉
オリジナル・キャラクタ「MEW」(みゅう)
(巻頭カラー写真の女の子です)



```
10 CLEAR 300,&HF1FF:GOSUB 300
20 COLOR 15, 7,4:SCREEN 2
30 A=0:D=VARPTR(A)
40 READ A
50 IF A=-1 THEN 50
60 B=A AND &H1F0
70 IF B=&HE0 THEN 100
80 IF B=&H1E0 THEN 150
90 GOTO 230
100 C=A-B
110 READ A
120 IF A=255 THEN 40
130 PAINT(PEEK(D+1),PEEK(D)),C
140 GOTO 110
150 C=A-B
160 READ A
170 B=PEEK(D):A=PEEK(D+1)
180 DRAW"bm=a;,=b;"
190 READ A
200 IF A=511 THEN 40
210 LINE-(PEEK(D+1),PEEK(D)),C
220 GOTO 190
230 POKE &HF204,PEEK(D+1)
240 POKE &HF205,PEEK(D)
250 READ A
260 POKE &HF202,PEEK(D+1)+&H20
270 POKE &HF208,PEEK(D)
280 A=USR(0)
290 GOTO 40
300 DEFINT A-Z
310 FOR A=&HF200 TO &HF23A
320 READ A$:POKE A,VAL("&H"+A$)
330 NEXT
340 DEF USR=&HF200
350 RETURN
360 REM --  ﾏｼﾝｺﾞ ﾃﾞｰﾀ  --
370 DATA 21,00,00,11,00,00,01,00,00,CD,4A,00,CD,22,F2,CD
380 DATA 22,F2,CD,4D,00,23,7B,07,07,07,07,5F,0B,78,B1,20
390 DATA E8,C9,F5,E6,0F,BA,20,0D,F1,D5,F5,7B,E6,0F,5F,F1
400 DATA E6,F0,B3,D1,F5,F1,07,07,07,07,C9
1000 REM ---  mew   DATA  ---
1010 DATA 484,-29952,30485,27686,25911,28752
1020 DATA 29587,-27501,-22651,-17284,-14981,-12929
1030 DATA -10113,-11133,-8829,-6529,-5509,-5506
1040 DATA -4230,-2705,-1953,-1974,-2509,-3297
1050 DATA -6134,-9728,511,228,-19923,255
```

```
1060 DATA 493,7756,9294,9806,14403,19515
1070 DATA 24886,27957,29751,32317,-31160,-30638
1080 DATA -30627,-30874,-30865,28271,23158,22409
1090 DATA 22666,20887,9629,8084,6528,5993
1100 DATA 6994,7756,511,237,17247,255
1110 DATA 16141,1550,493,-30864,30570,28266
1120 DATA 25199,23158,22409,-30845,-30864,511
1130 DATA 237,29565,255,-1011,3333,486
1140 DATA 57,3397,5450,8780,7770,7783
1150 DATA 7793,8062,8843,10133,11165,6557
1160 DATA 160,511,230.3446,255,489
1170 DATA 6220,5468,5494,6541,511,492
1180 DATA 24483,23708,22924,21135,18579,14488
1190 DATA 10902,9111,8866,11938,28068,25488
1200 DATA 22926,511,236,20893,255,-12788
1210 DATA 4356,493,156,2200,4760,7063
1220 DATA 10391,11680,11938,9891,8866,6820
1230 DATA 170,511,237,5024,255,27917
1240 DATA 4612,491,-30608,-29589,-28065,-27311
1250 DATA -25279,-22206,-17858,-16071,-16062,-17332
1260 DATA -20394,-18334,-16278,-13974,-11417,-12687
1270 DATA -15238,-16002,-12412,-8817,-7023,-5225
1280 DATA -3938,-5209,-6483,-8272,-10060,-10830
1290 DATA -12368,-15700,-23128,29096,23705,22929
1300 DATA 22922,24447,25979,28537,31355,-31103
1310 DATA -30608,511,235,-23696,255,490
1320 DATA -30603,-30590,-28030,-30603,511,234
1330 DATA -29571,255,490,-15237,-16766,-13690
1340 DATA -11125,-10609,-7280,-7285,-8316,-11132
1350 DATA -10879,-13441,-15236,511,234,-10104
1360 DATA 255,490,-10830,-14158,-16979,-21847
1370 DATA -25686,-30547,31405,27307,24996,23449
1380 DATA 26015,28834,-24153,-23642,-23129,-21857
1390 DATA -21097,-20318,-19544.-18527,-18007,-16730
1400 DATA -14686,-15191,-13138,-10321,-11036,511
1410 DATA 234,-14929,255,490,-28284,-22390
1420 DATA 511,490,-24157,-23405,-22390,-20094
1430 DATA -16766,511,490,-18301,-19828,-21097
1440 DATA 511,490,-18783,-18793,-18287,-16759
1450 DATA -15484,511,490,-13689,-15473,-16486
1460 DATA -16730,511,490,-14943,-13930,-13167
1470 DATA -9840,511,490,-8016,-9052,511
1480 DATA 490,-6994,-7777,511,490,-5718
1490 DATA -5989,511,-25078,3593,-1781,1553
1500 DATA 485,-20173,-23746,-26302,511,482
```

```
1510 DATA 23133,22901,511,489,32389,-29819
1520 DATA 511,481,-20658,-22199,-23737,511
1530 DATA 481,-21425,-23733,-21936,-23988,-21425
1540 DATA -23478,-21170,-21935,-23987,511,481
1550 DATA -24246,-24503,511,481,-16276,-14733
1560 DATA 511,481,-17039,-14987,-16528,-14733
1570 DATA -17042,-17040,-15243,-16787,-17552,-16012
1580 DATA 511,481,-22213,-19647,511,481
1590 DATA -13469,-11927,511,481,-19085,-19851
1600 DATA 511,495,-23176,-22664,511,486
1610 DATA -23173,-22661,511,486,31412,511
1620 DATA 495,-22408,-22920,511,495,-22867
1630 DATA 511,24836,17,5637,1539,-1
```



〈サンプルデータ 2〉
オリジナル・キャラクタ「アスキーねえちゃん」

```
10 CLEAR 300,&HF1FF:GOSUB 300
20 COLOR 15, 7,4:SCREEN 2
30 A=0:D=VARPTR(A)
40 READ A
50 IF A=-1 THEN 50
60 B=A AND &H1F0
70 IF B=&HE0 THEN 100
80 IF B=&H1E0 THEN 150
90 GOTO 230
100 C=A-B
110 READ A
120 IF A=255 THEN 40
130 PAINT(PEEK(D+1),PEEK(D)),C
140 GOTO 110
150 C=A-B
160 READ A
170 B=PEEK(D):A=PEEK(D+1)
180 DRAW"bm=a;,=b;"
190 READ A
200 IF A=511 THEN 40
210 LINE-(PEEK(D+1),PEEK(D)),C
220 GOTO 190
230 POKE &HF204,PEEK(D+1)
240 POKE &HF205,PEEK(D)
250 READ A
260 POKE &HF202,PEEK(D+1)+&H20
270 POKE &HF208,PEEK(D)
280 A=USR(0)
290 GOTO 40
300 DEFINT A-Z
310 FOR A=&HF200 TO &HF23A
320 READ A$:POKE A,VAL("&H"+A$)
330 NEXT
340 DEF USR=&HF200
350 RETURN
360 REM --  ﾏｼﾝｺﾞ ﾃﾞｰﾀ  --
370 DATA 21,00,00,11,00,00,01,00,00,CD,4A,00,CD,22,F2,CD
380 DATA 22,F2,CD,4D,00,23,7B,07,07,07,07,5F,0B,78,B1,20
390 DATA E8,C9,F5,E6,0F,BA,20,0D,F1,D5,F5,7B,E6,0F,5F,F1
400 DATA E6,F0,B3,D1.F5,F1,07,07,07,07,C9
```

```
1000 REM ---  asc1  DATA  ---
1010 DATA 481,-30343,-31623,32370,30825,29535
1020 DATA 27987,27221,27471,26451,27211,27203
1030 DATA 27703,28458,29217,29978,31760,-31479
1040 DATA -29435,-27389,-24829,-23548,-21502,-18943
1050 DATA -16125,-14070,-12782,-12006,-11999,-12240
1060 DATA -11973,-11199,-10171,-9655,-11190,-10164
1070 DATA -11695,-10159,-11945,-13727,-14998,-16266
1080 DATA -17794,-18049,-19075,-19591,-30599,511
1090 DATA 225,-28049,255,491,-30599,-31372
1100 DATA -32404,-32668,32343,32339,-31672,-30657
1110 DATA -30648,-29889,-29131,-28090,-27326,-27324
1120 DATA -26559,-26042,-25537,-24763,-24516,-23760
1130 DATA -22728,-21690,-20677,-20158,-19387,-17333
1140 DATA -15791,-16293,-17050,-17808,-18570,-19846
1150 DATA -20096,-20600,-20595,-18545,-19559,-19554
1160 DATA -19800,-30040,-30063,-29301.-28794,-28543
1170 DATA -30342,-30599,511,235,-28295,255
1180 DATA 481,-32398,-31111,511,487,-18812
1190 DATA -18822,-18566,-18556,-18306,511,495
1200 DATA 22463,23474,24487,25504,27030,28303
1210 DATA 30603,30859,32392,-30329,-29305,-29812
1220 DATA -29807,-28519,-26210,-23650,-22115,-20590
1230 DATA -18798,-16234,-14438,-13406,-12373,-11850
1240 DATA -11841,511,239,-13889,255,495
1250 DATA -30581,-29045,-29047,-31351,511,491
1260 DATA -28026,-29046,-29808,-29290,-28263,-25959
1270 DATA 511,235,-28781,255,494,-30073
1280 DATA -30836,-31086,-30568,-29282,-27741,-25947
1290 DATA -22619,-21600,511,494,30127,29375
1300 DATA 511,494,-15937,-16205,511,495
1310 DATA -29799,-29802,511,495,-29813,-29816
1320 DATA 511,481,-31409,-29364,-27316,-27315
1330 DATA -29618,-30894,-29361,-27825,-27058,-28593
1340 DATA -29617,-30894,-29875,-28083,511,481
1350 DATA -18099,-20662,-22195,-20146,-17584,-18868
1360 DATA 511,225,-20404,255,481,-30398
1370 DATA -27326,-27328,-31680,-31679,-27839,511
1380 DATA 481,-22462,-20159,-20160,-22463,-23230
1390 DATA -23229,-20926,511,490,-26032,-26282
1400 DATA -26786,-27038,-26266,-25242,511,490
1410 DATA -28544,-26747,-24953,-23420,-21887,-19845
1420 DATA 511,-24821,1295,490,-27538,-24977
1430 DATA -22931,-20631,-20632,511,490,-27794
1440 DATA -27797,511,28170,2313,492,-24641
1450 DATA -24653,-23895,-23392,-21608,-20080,-17784
```

```
1460 DATA -15484,-13441,-10111,-7546,-5234,-4197
1470 DATA -4437,-5193,-6721,511,492,-22849
1480 DATA -22857,-22353,-21083,-19812,-18281,-17007
1490 DATA -14195,-12150,-9585,-7788,-6245,-6234
1500 DATA -6736,-6735,-8005,-8769,511,236
1510 DATA -7745,255,494,-6991,-9793,511
1520 DATA 494,-12865,-6744,511,494,-6498
1530 DATA -15425,511,494,-19521,-8043,511
1540 DATA 494,-10097,-22337,511,494,-22350
1550 DATA -12916,-6745,511,494,-7504,-14961
1560 DATA 511,494,-16746,-8518,511,494
1570 DATA -10305,-18278,511,494,-19808,-12609
1580 DATA 511,494,-15169,-21080,511,494
1590 DATA -22094,-18497,511,494,-20801,-22343
1600 DATA 511,492,-6471,-6478,-6481,511
1610 DATA 492,-6506,-6509,511,492,-22110
1620 DATA -22102,511,484,5666,12800,15360
1630 DATA 15394,13858,13832,7969,5921,511
1640 DATA 484,10772,13844,13849,9753,511
1650 DATA 228,8473,11797,255,484,14410
1660 DATA 5450,5443,13123,14144,14139,13111
1670 DATA 6711,5683,5675,6437,15653,15658
1680 DATA 7978,7468,7471,7985,12849,14645
1690 DATA 15928,15936,15431,14921,13898,511
1700 DATA 228,14906,255,484,13428,7028
1710 DATA 6000,5974,6994,13394,15704,15710
1720 DATA 14174,14172,13401,8025,7770,7787
1730 DATA 8557,13677,14440,15976,15984,14964
1740 DATA 12148,511,228,12657,255,484
1750 DATA 10106,10132,12180,12154,10106,511
1760 DATA 228,10623,255,484,10141,10166
1770 DATA 12214,12189,10141,511,228,10918
1780 DATA 255,-8956,24,-2297,5,-2809
1790 DATA 1285,29959,2565,-7161,3844,5383
1800 DATA 4869,20749,5,24077,1285,18189
1810 DATA 2565,24333,3844,32525,4869,-1
```



K. ARAI

アニメC.G.に挑戦！

MSXポケットバンク [1]

1984年4月5日　初版発行　定価480円

著　者　川野名 勇・牧山慶士

発行者　塚本慶一郎

発行所　株式会社アスキー

〒107 港区南青山5－11－5住友南青山ビル5F

振　替　東京4-161144

電　話　03－486－7111（代表）

編集担当　八木淳一

表紙担当　郷　啓子

印刷　図書印刷株式会社

ISBN4-87148-733-4 C0055 ¥480E

MSX
POCKET BANK

MSXポケットバンク1

アニメC.G.に挑戦!

CONTENTS

ISBN4-87148-733-4 C0055 ¥480E

定価480円